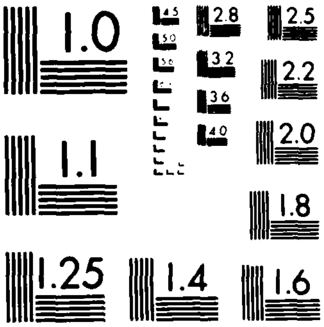
AD-A171 662 A COMPUTER SIMULATION OF DETONATION WITHIN AN ENERGETIC 1/1
MOLECULAR CRYSTAL(U) NAVAL ACADEMY ANNAPOLIS MD
A D BOYD 11 JUL 86 USNA-TSPR-134

UNCLASSIFIED F/G 9/4 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

17

AD-A171 662

A TRIDENT SCHOLAR PROJECT REPORT

NO. 134

A COMPUTER SIMULATION OF DETONATION WITHIN AN ENERGETIC MOLECULAR CRYSTAL

DTIC
ELECTE
SEP 8 1986
S D
B

UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND
1986

DTIC FILE COPY

86 9 8 056

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER U.S.N.A. - TSPR; no. 134 (1986)	2. GOVT ACCESSION NO. ADA161662	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A COMPUTER SIMULATION OF DETONATION WITHIN AN ENERGETIC MOLECULAR CRYSTAL.		5. TYPE OF REPORT & PERIOD COVERED Final: 1985/1986
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Boyd, Alan Davis		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS United States Naval Academy Annapolis.		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS United States Naval Academy, Annapolis.		12. REPORT DATE 11 July 1986
		13. NUMBER OF PAGES 79
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release; its distribution is UNLIMITED.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

This document has been approved for public release; its distribution is UNLIMITED.

18. SUPPLEMENTARY NOTES

Accepted by the U.S. Trident Scholar Committee.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Detonation
explosions
crystals

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The study of detonation has been based upon hydrodynamic theory. This view of detonation completely ignores the actual chemistry of the explosive reaction. Therefore, the dynamics of detonation on a molecular level remain unknown. The purpose of this project is to use a computer to investigate the propagation of detonation through a crystal.

Research in this area is hindered by the fact that monitoring

(OVER)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) instruments are destroyed in actual detonations. Computer simulations avoid this problem because there is no physical explosion. The actual detonation is extremely rapid; collection of data at designated conditions or times can not be guaranteed. The computer does not have this problem since it can be programmed to display the data at any desired condition or time.

✓

A-1

U.S.N.A. - Trident Scholar project report; no. 134 (1986)

"A Computer Simulation of Detonation within an Energetic Molecular Crystal"

A Trident Scholar Project Report

by

Midshipman Alan Davis Boyd, Class of 1986

U. S. Naval Academy

Annapolis, Maryland

Associate Professor Mark L. Elert
Department of Chemistry
B.S., Michigan State University
Ph.D., University of California, Berkeley

Accepted for the Trident Scholar Committee

Carl Schneider

Chairman

11 July 1986

Date

ABSTRACT

The study of detonation has been based upon hydrodynamic theory. This view of detonation completely ignores the actual chemistry of the explosive reaction. Therefore, the dynamics of detonation on a molecular level remain unknown. The purpose of this project is to use a computer to investigate the propagation of detonation through a crystal.

Research in this area is hindered by the fact that monitoring instruments are destroyed in actual detonations. Computer simulations avoid this problem because there is no physical explosion. The actual detonation is extremely rapid; collection of data at designated conditions or times can not be guaranteed. The computer does not have this problem since it can be programmed to display the data at any desired condition or time.

A nonhomogeneous crystal of diatomic molecules was monitored to discover the atomic interactions during detonation. A Lennard-Jones potential equation was used to represent the exothermic reaction between diatomic hydrogen and chlorine molecules. This is the first project to use the natural formation of stable reaction products to achieve exothermicity.

Table of Contents

Computer simulations are increasingly used in scientific research. They can be used to verify current theory by reproducing laboratory results or to imitate a system which can not be empirically monitored in the laboratory. The second use is extremely beneficial because one can write a computer program which generates output at any given time or condition. The computer simulations can isolate particular parameters and monitor the effects of various changes to these parameters. The computer simulation is also very versatile, since it can focus in on either macroscopic or microscopic iews of a system. The main advantage of a computer simulation is the number of computations which can be made within a given amount of time.

My project, "A Computer Simulation of Detonation within an Energetic Molecular Crystal," took advantage of the computer's ability to generate and monitor a system which can not be observed through ordinary laboratory means. Problems with monitoring detonation in the laboratory are that any monitoring devices are usually destroyed and the speed of the reaction prevents accurate assimilation of data. The computer simulation is obviously not explosive in physical terms; therefore, the first problem of destroying the monitoring equipment is eliminated. The problem of the reaction's speed is easily avoided because the computer program can be written to display the detonation at any given time or condition during the reaction. The data generated by the computer can very easily be stored in ordered files for future use. The data stored is also selected by the programer. He can generate tables of numbers or even graphical output. These advantages make computer simulations a very good medium for studying detonation--if the proper equations relating the interactions between the atoms can be found. The main focus of my project was to see if a program could solve the simultaneous differential equations for a large number of particles within a reasonable amount of computer run time.

Previous Works

Scientists have done studies of shock waves in earlier simulations. A. M. Karo, J. R. Hardy, and F. E. Walker did work with a homogeneous monoatomic crystal.[1] Their work was on shock induced detonation. They used Morse potentials for interaction between atoms, but the atoms only interacted with its first and second neighbors in the crystal. If the crystal array is disrupted, the atoms "become closely juxtaposed without 'sensing' one another's presence."[2] Although this greatly reduces the run time of the program, it allows atoms not normally next to each other to "pass through" one another. This does not pose any major problems when low values of initial velocity are given to the crystal, but when larger values of velocity or random motion perpendicular to the shock wave propagation are used the crystal structure disintegrates.[3]

Karo, Hardy, and Walker also observed that part of the crystal departed the main lattice structure when a detonation wave impinged a free surface.[4] These problems can be solved if they allow each atom to interact with every other atom in the crystal and if they reduce the number of atoms located on a free surface in the crystal being observed.

Two types of potential equations were used: endothermic and exothermic. These equations were used during separate computer runs. The endothermic potential equation represented the breaking of bonds between adjacent atoms. The runs made with this potential function had a quiescent behavior. The exothermic potential equation was written by the authors to be an exothermic response to a bond stretched beyond a certain limit. This reaction does not occur in nature. The entire lattice was torn apart when this unnatural exothermic equation was used.[5] These equations are fine for initial research but they are not the results of a natural detonation. The energy from detonation comes from atoms, molecules, and molecular fragments forming more stable molecules after the detonation wave has passed. The energy released by the more stable molecules is what propagates the detonation wave and causes the thermal and shock effects of the explosion.

D. H Tsai and S. F. Trevino did studies on shock wave propagation through a homogeneous diatomic crystal. This work is closer to actual detonation, because their hypothetical diatomic molecules can break apart and remain unattached or they can reform diatomic molecules. The reformed molecule is the same molecule which was originally present; therefore more stable substances are not produced. A potential energy equation which is nonoccurring in nature must be used to make the equation exothermic. They used two different Morse potential equations: one for dissociated atoms and one for atoms bonded together as a diatomic molecule. The dissociation of the atoms was given an exothermicity factor, like Karo, Hardy, and Walker used, which does not occur in nature. Each atom had a flag to show whether or not it was bonded to another atom. The value of this flag would determine which of the two potential energy equations would be used. Figures 1 and 2 are of the intramolecular and intermolecular potentials respectively. Note the strong intermolecular potential in Figure 2; normally intramolecular forces are stronger than intermolecular ones. Figure 3 shows the endothermic and exothermic potential curves used by Tsai and Trevino. The upper curve is the intramolecular potential; it does not take much activation energy to get it to reach the intermolecular curve. Once an atom is "influenced" by the intermolecular curve, it will dissociate to become stable. The stability gives the reaction energy, but as mentioned before, it is not a natural phenomenon.[6]

R. A. MacDonald and D. H. Tsai worked on dynamical

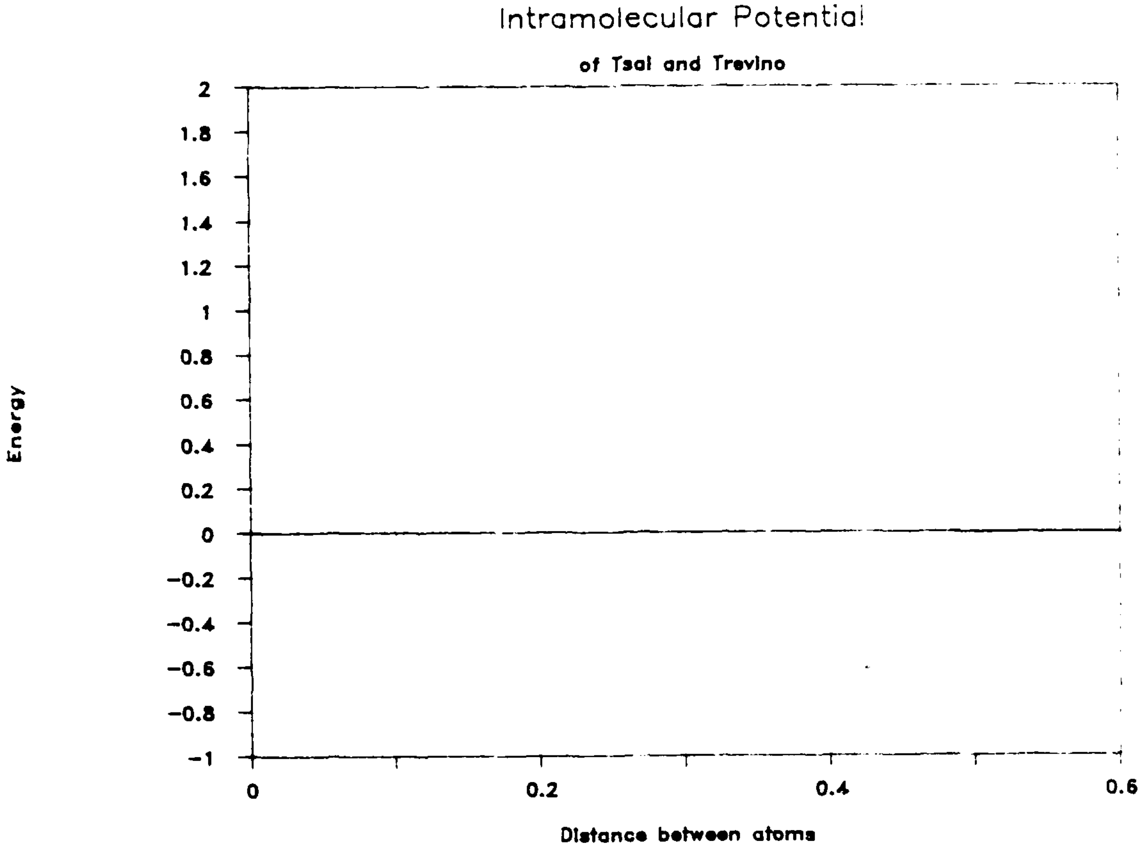


Figure 1

Intramolecular potential equation invented by Tsai and Trevino

Energy units are arbirary.

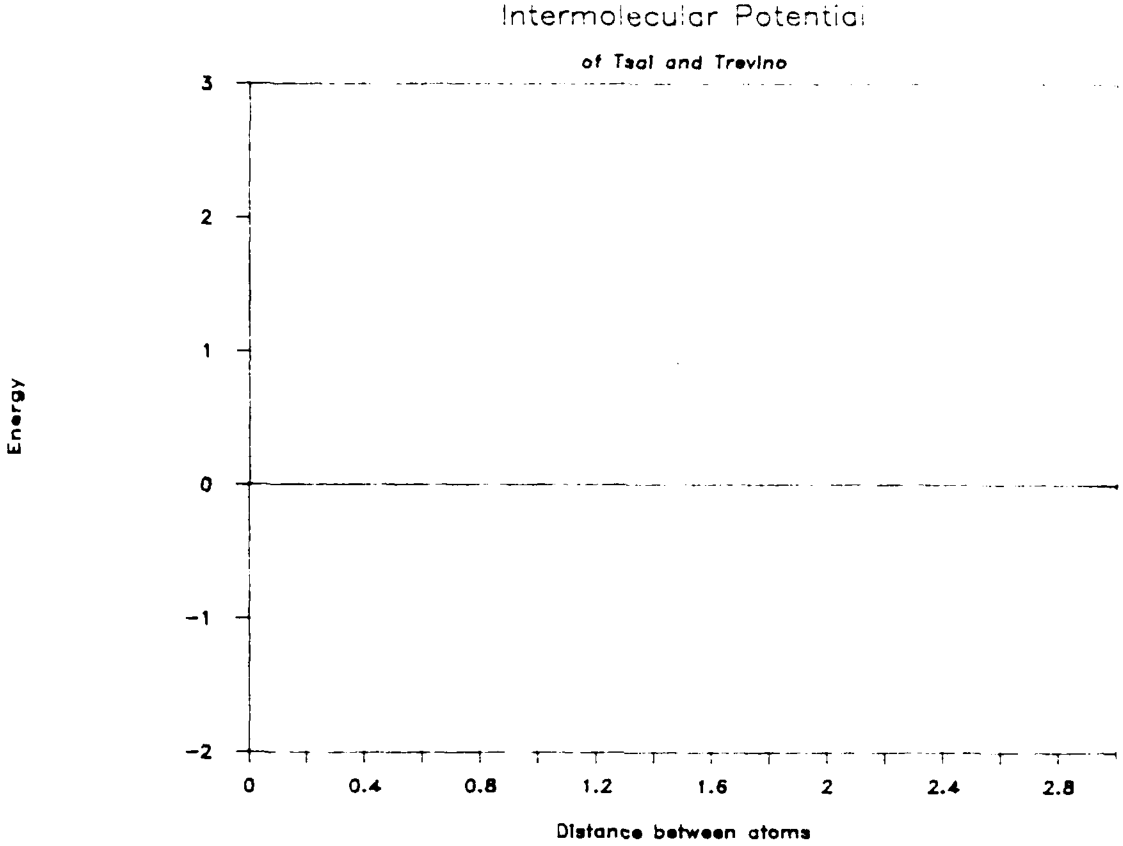


Figure 2

Intermolecular potential equation invented by Tsai and Trevino. Energy units are arbitrary.

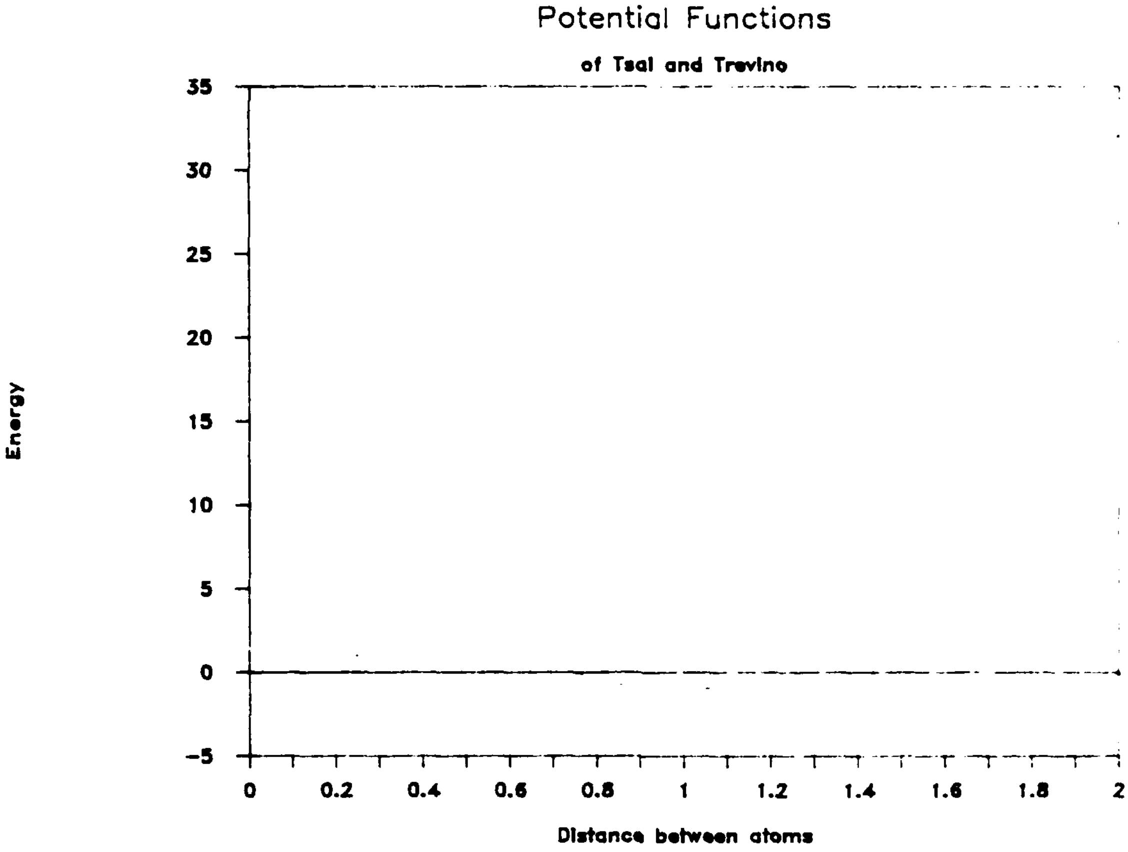


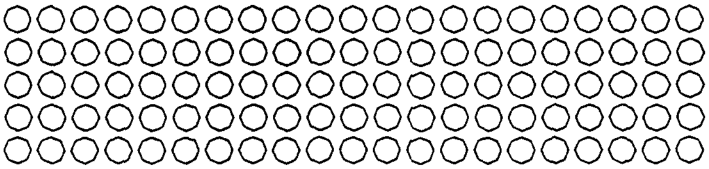
Figure 3

The change in potential from intramolecular to intermolecular interactions. The energy units are arbitrary.

calculations of energy transport in crystalline solids. Their study was testing whether or not thermal equilibrium existed behind the shock front in a solid. They concluded the thermal equilibrium propagated at such a slow velocity that thermal equilibrium was not achieved behind the shock wave. The computed results showed that a relaxation region existed behind the shock wave, but that it never reached equilibrium. The kinetic and potential energies would not transmit the equilibrium fast enough to follow the shock wave. Their results did show that the relaxation time is dependent upon the internal degrees of freedom of the molecule. They originally only had included coupling between molecules and not intramolecular degrees of freedom. When the intramolecular degrees of freedom were included, the relaxation time was shortened.[7]

Initial Conditions

To avoid some of the problems encountered by the other scientists, I utilized several different initial conditions. These differences included original atomic spacing, periodic boundaries, expansion of the crystal, and interaction between all atoms. The crystal was not set up starting at the origin. Instead the crystal was shifted out along the x-axis 15 Angstroms and shifted up one half of the equilibrium spacing between molecules along the y-axis. (Figure 4) The reason for moving out in the x direction is to allow the atoms receiving initial momentum and velocity to approach the



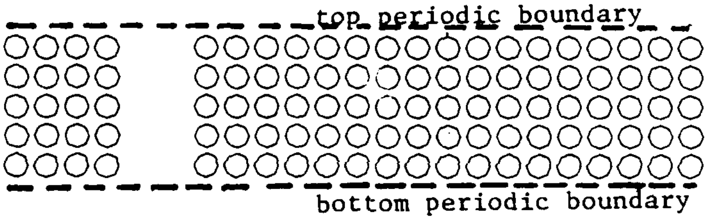
Monoatomic crystal generated by computer in perfect rows and columns. Note that monoatomic molecules are used for simplicity of showing the initial parameters.

Figure 4

crystal with natural interactions. It is not correct for matter to instantaneously change states. (Even capacitors take a small amount of time to charge up in electronic circuitry). The rest of the crystal is also allowed to settle into equilibrium position before the detonation is initiated. The shift in the y direction is related to the periodic boundaries and the expansion of the crystal.

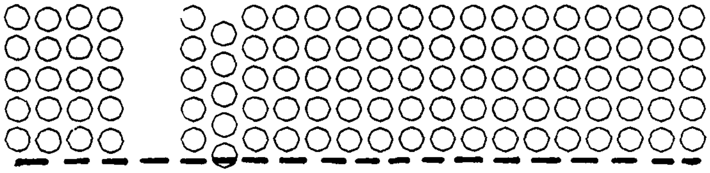
Periodic boundaries will eliminate the problem of having large numbers of observed atoms along free surfaces. The top and bottom edges make up the majority of the crystal's free surface area. Approximately forty percent of all atoms involved in the simulation are situated along the edge. This would be an unrealistic ratio in an actual physical crystal. Physical crystals have most of their atoms on the inside, while only a small ratio of the atoms are located on a free surface. There are two ways to solve this problem: use several million atoms in the simulation or expand the crystal. The use of millions of atoms would make the program's run time on the order of years instead of days at the current state of computer technology. Using that large a number of atoms in a simulation is not feasible. The use of periodic boundaries only doubles the number of atoms involved in the simulation, and significantly lowers the ratio of atoms along a free surface to approximately four percent. Periodic boundaries were set up to keep the atoms contained in the crystalline area. This avoids the problem experienced by Karo, Hardy, and Walker in which the atoms along a free surface departed the crystal structure into free space. My periodic boundaries were set at one half the equilibrium distance between molecules in the y direction. This shifts the crystal up one half the equilibrium distance along the y-axis. (Figure 5) The other periodic boundary exists one half the equilibrium distance above the center of the initial position of the top row of atoms. These boundaries remain in the same position throughout the computer run. If an atom ever crosses a boundary, (Figure 6), then it is shifted to the other periodic boundary. (Figure 7) The shifted atom only changes position in the y-coordinate; the x-coordinate, the velocity, the momentum, and the momentum derivative remain the same. If the atom in Figure 6 has a y-cooridinate .002 Angstroms below the periodic boundary at the y-origin, then its y-coordinate is shifted to .002 Angstroms below the periodic boundary on the top. This effectively wraps the crystal around along the y-cooridinates. (Figure 8) The expansion of the crystal is an extension of the periodic boundaries; the expansion enables the movement of atoms from one periodic boundary to the other to be a natural occurrence.

Expanding the crystal involves generating a new set of atoms by taking atoms with y-coordinates situated between the origin and one half of the top periodic boundary, (Figure 9), and shifting them up above the top periodic boundary.



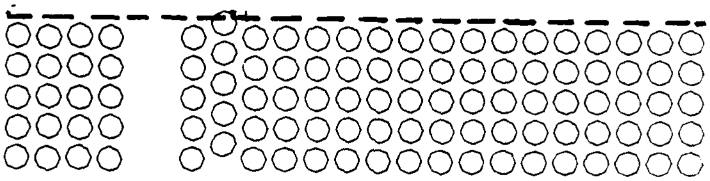
Positions of the periodic boundaries around the crystal

Figure 15



Atom crossing lower periodic boundary

Figure 6



Atom shifted to top of crystal after it has crossed periodic boundary

Figure 7

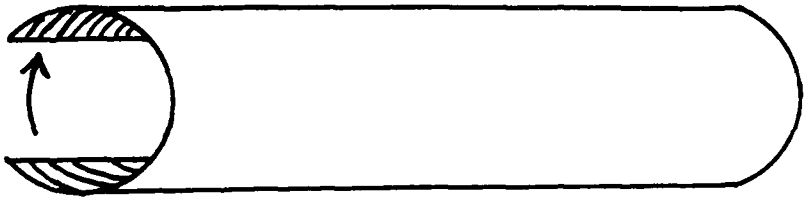
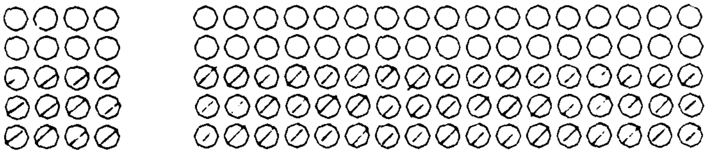


Figure 8

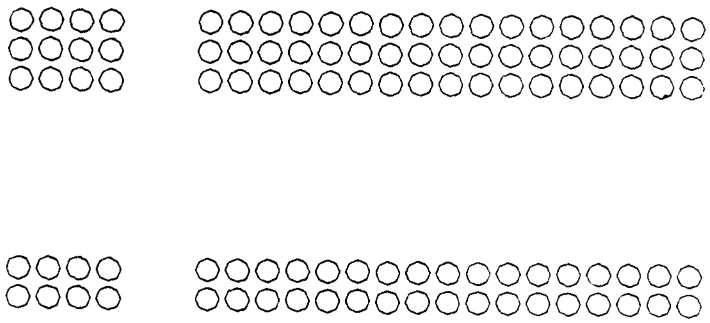
The physical effect of wrapping the crystal's edges



Atoms with shading will be projected abouve the crystal during "crystal expansion."

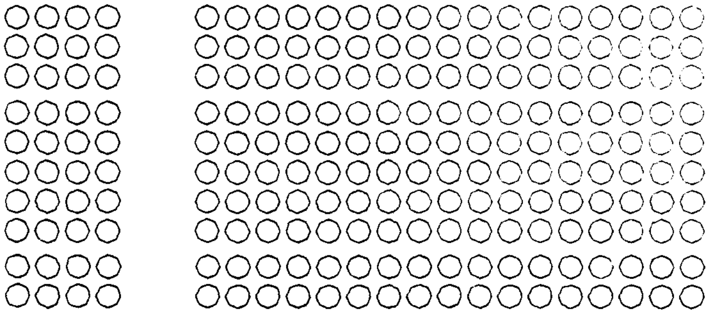
Figure 9

(Figure 10) Then the remaining atoms, those in the top half of the original crystal, are shifted below the bottom periodic boundary. This has doubled the number of atoms involved in the simulation. (Figure 11) The new atoms are contained only in temporary memory, and they are regenerated during each time step. They only have positions both x and y; they do not have their own velocity, momentum, or momentum derivative. These parameters are not needed since the new atoms are projected images of the original atoms. The velocity, momentum, and momentum derivatives from the original atoms will correctly move the new expanded atoms. The reason for having the new expanded atoms is to enclose the original crystal. The atoms from the original crystal are now interior atoms, which will give more accurate predictions than atoms impinging a free surface edge. Looking back at Figure 6, there is a "hole" generated above the column which is shifted down. By expanding the crystal this hole has been filled by an expanded atom since the beginning of the program. Now that the original atom has crossed the periodic boundary, the original atom is moved into this hole. One might think that this movement creates a new hole where the original atom was moved from. This is not the case because the expanded crystal now generates an atom in this hole during subsequent time steps. The expanded atoms, therefore, are actively interacting with the original atoms during each time step.



Projected atoms generated during crystal expansion

Figure 10



Crystal generated after expansion. Note that the space in between the new atoms and the original crystal was artificially put in the figure so that the difference is easily noticed.

Figure 11

Potential Equations

The interaction between the atoms is caused by the potential equations. Two potential equations are used in computer simulations: the Morse equation and the Lennard-Jones equation. The two are very similar--except for the behavior of the atoms at small interatomic distances. The Morse equation has a finite potential value when the interatomic distance goes to zero. This accounts for atoms "passing through" each other which is physically impossible. Karo, Hardy, and Walker had stated that the phenomenon of atoms passing through one another occurred during their simulations. They had used the Morse potential equation which contributed to this error. The Lennard-Jones potential equation goes to infinity as the interatomic distance goes to zero. (Figures 12-14) Atoms experience this repulsion in real life; therefore, the Lennard=Jones equation is more accurate when close interatomic distances will be experienced. The Lennard-Jones equation is normally given in the form:

$$U(r) = 4\epsilon * ((\sigma/r)^{12} - (\sigma/r)^{6})$$

r - inter-atomic distance

Lennard-Jones potential equations are normally intermolecular. For this simulation the parameters were changed to have intramolecular interactions. This was done to allow an atom to interact within the original molecule it was

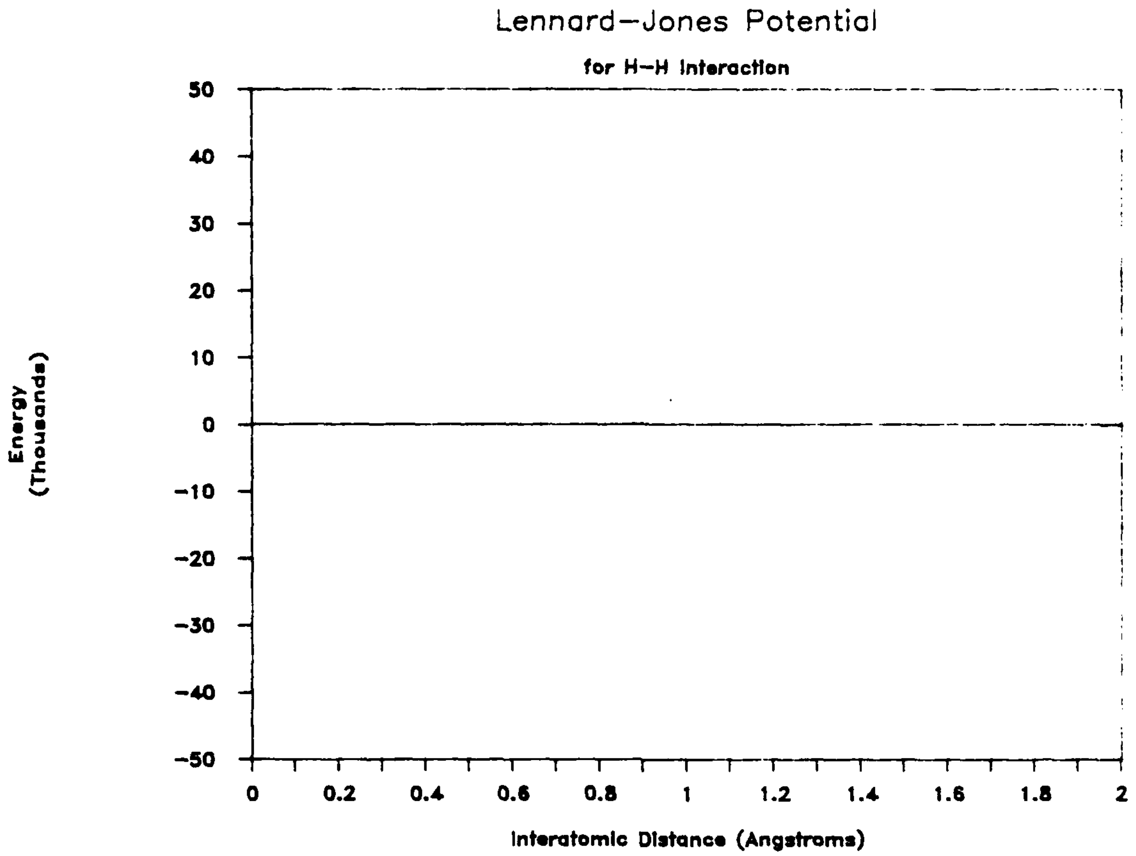


Figure 12

Lennard-Jones potential for interatomic H-H interactions. The energy units are amu $\mathring{A}^2$/ $psec^2$.

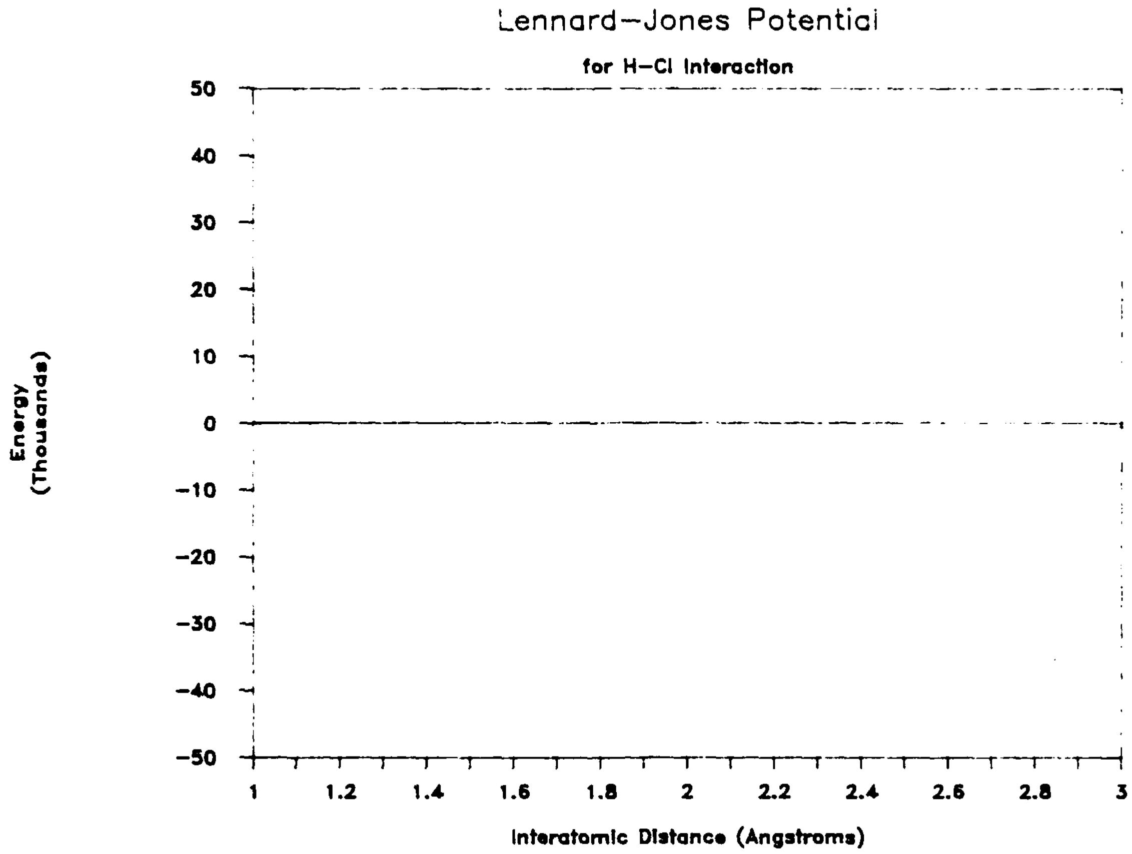


Figure 13

Lennard-Jones potential for interatomic H-Cl interactions. The energy units are amu $\AA^2$/ $psec^2$.

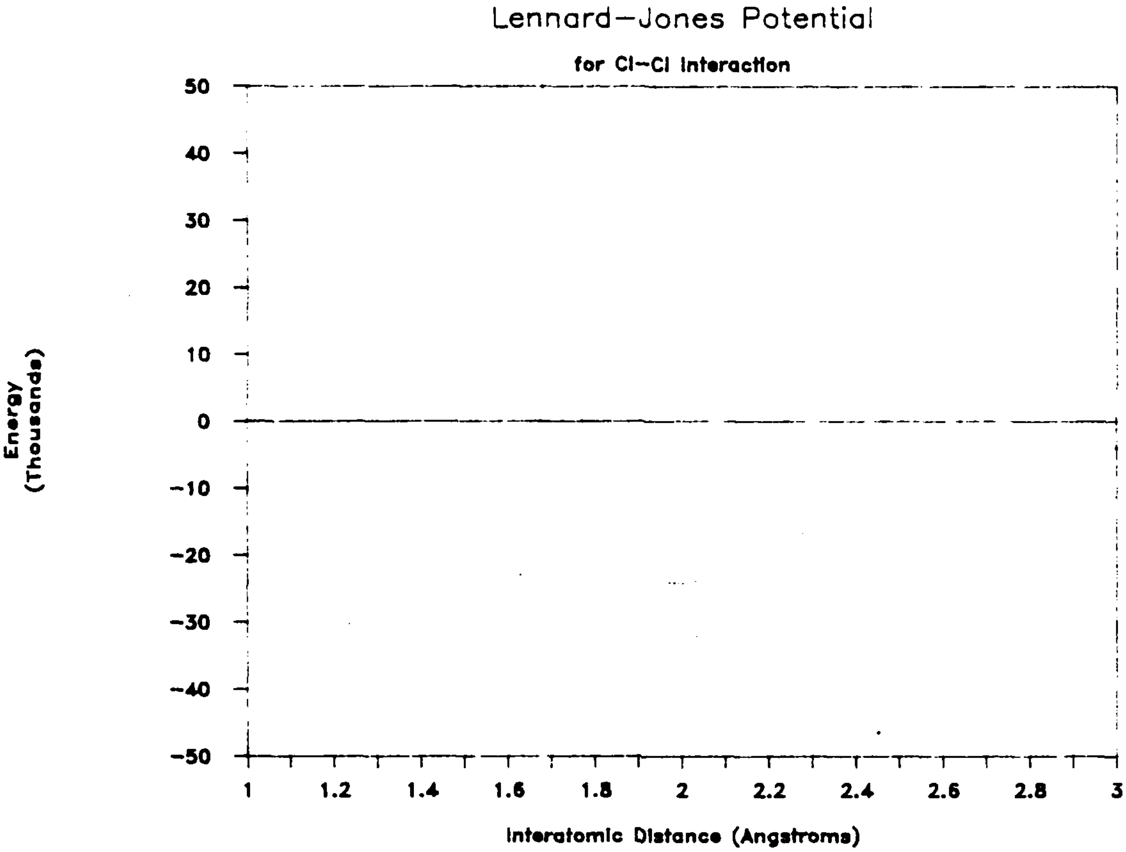


Figure 14

Lennard-Jones potential for interatomic Cl-Cl interactions. The energy units are amu Å^2/ psec^2.

assigned to, while being able to interact with the surrounding atoms to form the most stable compound possible. This was simply done by taking the equilibrium intramolecular distance for each of the three molecules, HH, HCl, and ClCl, and solving for the constants o and . The equilibrium interatomic distance is related to the o term by a multiple of $2^{1/6}$. The term is a measure of the extent of attraction between pairs of molecules. The values for the attraction was taken from Molecular Theory of Gases and Liquids by J. O. Hirschfelder, C. F. Curtiss, and R. B. Bird.[8] Units were converted to a system of Angstroms, picoseconds, and atomic mass units to avoid the use of large exponents with the potential energy values. This system also made the position values manageable numbers with the Angstrom unit measure. Lennard-Jones potential equations are very accurate for non-polar molecules, HH and ClCl; however for polar molecules, HCl, the equations are not as accurate. The Lennard-Jones potential "may be useful for purposes of calculations until the theory needed for describing complex molecules has been developed."[9]

The Lennard-Jones potential equations; therefore, are best for the requirements of my simulation. To save calculation time the form of the equation was converted to:

$$U(r) = B * (1 / r)^{12} - A * (1 / r)^{6}$$

The constants 4, , and o were raised to the appropriate power and then multiplied together. This alteration saves six calculations per atom-atom interaction. The interatomic distance raised to the twelth power causes the repulsion between atoms at close distances. (Figure 12) The negative of the sixth power of the interatomic distance produces the bottom of the well, so that there is an equilibrium distance between the atoms.[10]

Equations of Motion

The equations of motion are the momenta and coordinate equations for each atom. These exist in all three coordinates, but my computer simulation is limited to two dimensions. The Hamiltonian equations are:

$$\begin{aligned}\dot{q}_k &= \partial H / \partial p_k \\ &= \partial T / \partial p_k \\ \dot{p}_k &= -\partial H / \partial q_k \\ &= -\partial V / \partial q_k\end{aligned}$$

During each time step of the simulation, each atom has a set of these equations with every other atom in the expanded crystal. The momenta and coordinate equations are directly dependent upon one another; they are simultaneous differential equations. The kinetic energy, T, is equal to the summation of the momentum squared divided by the twice the mass of the atom. The V term, the potential energy, is the Lennard-Jones equation. The identity:

$$\partial f / \partial q_k = (\partial f/\partial r)(\partial r/\partial q_k)$$

is used to solve the momenta equations, since the Lennard-Jones equation must be solved with respect to the coordinate derivative. The equations are:

$$T = p_k^2 / (2 * m_k)$$

$$\dot{q}_{ik} = p_{ik} / m_k$$

$$V = B * r^{-12} - A * r^{-6}$$

$$r_{jk} = (q_{ij} - q_{ik})^{2} \quad {}^{1/2}$$

$$\dot{p}_{ik} = -dV / dq_{ik}$$

$$= (-dV / dr)(dr / dq_{ij})$$

subscript i refers to x or y-coordinate

These equations are in a usable form for solving the atomic interactions. Substituting in the actual Lennard-Jones equation for $\dot{p}_{ik}$ gives the equation used in the subroutine which calculates the force on each atom. This equation is:

$$\dot{p}_{ik} = (12*B*r^{-14} + 6*A*r^{-8}) * (q_{1k} - q_{2k}) \; (-1)^{i+1}$$

These equations are now ready to be integrated to calculate the momenta and coordinates for each atom.

Integration Techniques

Now let us address the problem of properly integrating the equations of motion and momentum. The main problem is that the program has several simultaneous differential equations for each atom. This requires five hundred seventy-six sets of simultaneous equations to be solved during each time step. Many pre-written integration programs, such as DGEAR from the International Mathematics and Science Library, IMSL, can not efficiently handle such a large number of simultaneous equations. Research into integration techniques determined that the Euler methods might be work. Unfortunately the Euler method generates an error in the calculation of a new position, X_{n+1}, if the velocity is not constant over the time interval. The Euler method is a most simple method, but it lacks accuracy. An extremely small step size is required to get any accuracy.[11] In real life the atoms interact at infinitely small time intervals, i.e. continuous integration. Computer simulations require that a reasonable time interval be selected; since the forces acting upon each atom changes with its position, the velocity is constantly changing. This generates an error over time increments in which the velocity is not constant. Each integration with the Euler method has an error. The error starts out small, but each subsequent integration increases the error from the correct value. The Euler method becomes more accurate the smaller the time increment, but the error generated by a time increment which has a reasonable program run time is unacceptably large. Fortunately there is a Modified Euler method which is over twenty-two times more accurate for each integration over the same time increment.[12] It should be noted that as the time increment becomes infinitely small, both the Euler method and the Modified Euler method will give the same correct answer. The Modified Euler method uses the arithmetic average of the velocities at the beginning and end of the time increment. This average velocity more accurately predicts the true position. The equation for the Modified Euler method is:

$$X_{n+1} = X_n + .5 * (V_n + V_{n+1}) * h$$

X - position V - velocity h - time increment

One can not use the Modified Euler method directly from the equation. The equation requires that V_{n+1} be known to predict the value of X_{n+1}. The Euler method is used to predict a value for X_{n+1}; this value is temporarily stored to calculate a value for V_{n+1}. This value of V_{n+1} is then used in the Modified Euler equation to predict an accurate value for X_{n+1}. Other integration methods exist, example the Runge-Kutta method, but their run time makes them inefficient for this particular problem. The Runge-Kutta method is a fourth order solution, which doubles the run time and storage required for the program.

The Program

The actual program is included in Appendix A. It is written in Fortran 77. The program starts out by dimensioning all of the arrays required for the program. The position, velocity, momentum, and momentum derivative are set up for the x and y terms for each atom in the crystal. The x and y position for the expanded crystal are also included. A second set of arrays for position, velocity, momentum, and momentum derivative has been set up to use the Modified Euler integration method. The final array dimensioned is to record the type (T) of the atom. The parameters for the crystal size follow the arrays. The LENGTH is a summation of the equilibrium distance between two hydrogen atoms, a HH molecule and a ClCl molecule, two chlorine atoms, and a HH molecule and a ClCl molecule again. This is the periodic cycle to set up the initial crystal. The equilibrium distance between a HH molecule and a ClCl molecule was received from A. Blumen and C. Merkel in their ab initio study of hydrogen chloride.[13] The YMOVE term is the top periodic boundary.

The opening of external files is the next major portion of the program. The 'totena' file stores the total energy of the crystal; the total energy of the crystal is periodically calculated to monitor the validity of the simulation. If the simulation is bad, then the total energy will exponentially increase. The 'savfla' file saves the type, position, and momentum of each atom at designated times, so that if the computer crashes the program can be continued without restarting from the beginning. The 'ta' and 'ia' files store stop action frames of the crystal. These files are generated to display the type, size, and position of each of the original atoms in the crystal.

The subroutines which set up the initial crystal and give initial momentum to selected atoms are called. These subroutines are towards the end of the program. The POSition subroutine starts on page A-9. The initial positions are constructed by the use of two Do loops; one is used for rows and the other is for columns. As each atom is assigned a position, it is also assigned a Type. The rows alternate which type of atom is on the left side of the crystal. The alternation is flagged by the value of the 'J' variable. To vertically align the center of each of the diatomic molecules the 'DISTCENT' term is used on every other row. Finally the first four atoms in each row are moved back to allow the crystal to interact naturally.

The MOMentum subroutine is located on page A-7. This subroutine sets all initial momenta to zero; then it gives the first four atoms in each row initial momentum. The hydrogen and chlorine atoms get different initial momentum values. The values of the momentum in the x direction are calculated so that the atoms get the same initial velocity. The momentum in the y direction is randomly generated to simulate random thermal energy.

A comment block stating the units, back in the main program, follows the return from the MOM subroutine. The initial time, time increment, and final time are now set. These three terms are used in a Do loop, which contains the integration and prediction portions of the Modified Euler method. The initial positions of the atoms are sent to a file through the WRITFILE subroutine. The WRITFILE subroutine is on page A-11. This subroutine generates a file which displays the positions of the atoms in the crystal. An abridged output file is included in Appendix A. The middle atoms were omitted from the appendix to keep the data at a reasonable level. The data file contains commands for the PS300 computer to display and annotate the crystal. Each atom must be stored as a list of vectors, so that it can be displayed in a circular form. The atom type determines which color the display is. Hydrogen atoms are red and chlorine are green; these are the standard colors used to represent these two atoms. The data stored by the WRITFILE subroutine will be used to animate the molecular interactions.

The LJ subroutine is the next major portion of the program. LJ stands for Lennard-Jones, because this subroutine calculates the Lennard-Jones potential between atoms. The LJ subroutine is listed on page A-5. The momentum derivatives are reset to zero, so that the summation of each atom's total potential energy is only for the current time step. The velocity is then calculated by dividing each atom's momentum by the appropriate mass. Expansion of the crystal is the next major routine. This is done by comparing the y-coordinate position with the center of the crystal. If the atom is above the center of the crystal, then another atom is generated beneath the bottom periodic boundary the same distance the original atom was beneath the top periodic boundary. Both of these atoms have the same x-coordinate value. If the original atom is below the center of the crystal, then an expanded crystal atom is generated above the top periodic boundary. Now the subroutine AL is called to calculate the potential forces between the atoms.

Subroutine AL simply calculates the potential force for each atom in the original crystal. The distance between the atoms in both the x and y-coordinates is calculated, then these two terms are combined to find the magnitude of the vector distance between the two atoms. This distance is then used in the derivative of the Lennard-Jones potential energy equation to calculate the force between the atoms. The force is stored in the momentum derivative array. The computer determines whether the atoms are two hydrogens, two chlorines, or a combination so that the proper coefficients may be used. The direction of the force is determined by the magnitude, positive or negative, of the x and y distances. The subroutine moves on to the next atom of the original crystal when it has calculated the force between this atom and all other atoms within the expanded crystal. It has been determined that interatomic distances of greater than ten Angstroms will not generate a significant force. Therefore, if the magnitude of the vector distance is greater than ten, then the calculation of the force is skipped to save run time. After all of the original atoms have had their new forces calculated the AL subroutine returns to the LJ subroutine. The LJ subroutine immediately returns to the main program to be used in the Modified Euler integration method.

The velocity and momentum derivative values returned from the LJ subroutine are used to predict the position and momentum values by the Euler method. These new values are sent back to the LJ subroutine to calculate a new set of velocities and momentum derivatives. The LJ subroutine repeats the force calculation process, and returns the second set of velocities and momentum derivatives. The first and second sets of velocities and momentum drivatives are used in the Modified Euler integration method to calculate the correct new positions and momenta.

The program now checks to see if the current time is one of the designated output flags. If it is then an output file is generated, the total energy of the system is calculated, and the important parameter data is stored. Finally the original atoms are checked to see if they have crossed a periodic boundary. If one has, it is moved to its appropriate position near the other periodic boundary. This completes one pass for the computer run; the program will now loop through this routine until the final time is reached.

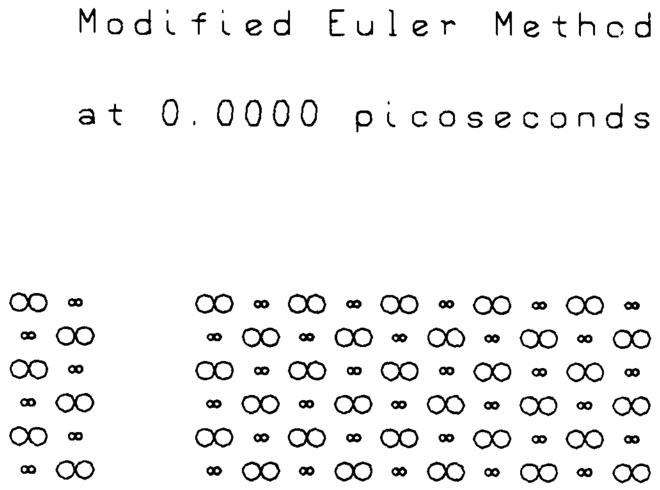
Conclusions

The program is the first computer simulation of detonation using either a realistic exothermic reaction or nonhomogeneous diatomic molecules. This is a major step towards simulations of explosives which are in current arsenals. These molecules are too large to have a computer simulation with present technology. I used the Lennard-Jones potential equation for a hydrogen and chlorine crystal. (Figure 15) The original crystal consisted of HH and ClCl molecules, as the detonation progressed the molecules separated and began to form HCl. (Figure 16)

The run time on the DEC VAX 11/780 computer is approximately six days to simulate one picosecond. This is actual working CPU time--not elapsed clock time. The speed of the VAX is one million instructions per second, 1 mips. A supercomputer with the speed of twenty mips could easily handle this program. Because of the length of run time and the periodic computer crashes, the amount of data I could obtain is limited.

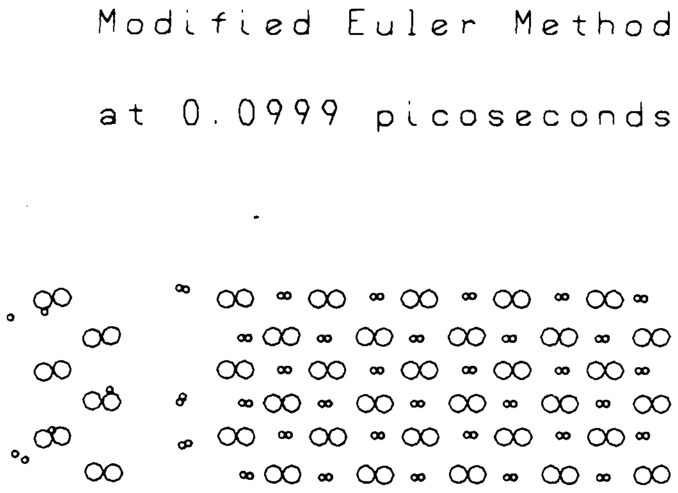
The data for each simulation run is collected in data files; an example is shown on pages A-14 to A-20. These files display a color representation of the crystal. The crystal is two dimensional and the display accurately shows the progression of the detonation wave. The only drawback to my program is that it allows the atoms to clump together. Since this is only a simulation, the clumping factor is not very detrimental. In fact it increases the exothermicity of the reaction.

Future work made possible by my project will explore different parameters of detonation. Changing the crystal orientation, different exothermicity values of reaction,



Initial hydrogen and chlorine crystal

Figure 15



Hydrogen and chlorine crystal generated during a computer run.

Figure 16

larger molecules, and the directivity of the crystal shape are examples. Ideally computer simulations will be used to invent more effective explosives. Knowledge of detonation and its effects is leading to this predictive capability.

Endnotes

[1] A. M. Karo, J. R. Hardy, and F. E. Walker. "Theoretical Studies of Shock-initiated Detonations," <u>Acta Astronautica</u> Vol. 5, p. 1041.

[2] Ibid., p. 1044.

[3] Ibid., p. 1048.

[4] Ibid., p. 1049.

[5] Ibid., p. 1050.

[6] D. H. Tsai and S. F. Trevino. "Simulation of the Initiation of Detonation in an Energetic Molecular Crystal," J. Chem Phys 81 (12) pt I., 15 Dec 84, pp. 5636-5637.

[7] R. A. MacDonald and D. H. Tsai. "Molecular Dynamical Calculations of Energy Transport in Crystalline Solids," North Holland Publishing Co. Amsterdam.

[8] J. O. Hirschfelder, C. F. Curtiss, and R. B. Bird. <u>Molecular Theory of Gases and Liquids</u>, John Wiley & Sons, Inc., New York, 1954, p. 1110.

[9] Ibid., p. 1110.

[10] Gordon Barrow. <u>Physical Chemistry</u>, 4th ed., McGraw-Hill Book Co., New York, 1979, p. 10.

[11] Curtis F. Gerald. <u>Applied Numerical Analysis</u>, 2nd ed., Addison-Wesley Publishing Company, Reading, Mass., 1980, p. 255.

[12] Ibid., p. 257.

[13] A. Blumen and C. Merkel. "An <u>ab initio</u> study of hydrogen chloride," Phys. B: Atom Molecule Phys., Vol 10, No. 15, 1977, p. L555.

Bibliography

Barrow, Gordon. Physical Chemistry, 4th ed., McGraw-Hill Book Co., New York, 1979.

Blumen, A. and C. Merkel. "An ab initio study of hydrogen chloride," Phys. B: Atom Molecule Phys., Vol 10, No. 15, 1977, pp. L555-556.

Gerald, Curtis F. Applied Numerical Analysis, 2nd ed., Addison-Wesley Publishing Company, Reading, Mass., 1980.

Hirschfelder, J. O.; C. F. Curtiss; and R. B. Bird. Molecular Theory of Gases and Liquids, John Wiley & Sons, Inc., New York, 1954.

Karo, A. M.; J. R. Hardy; and F. E. Walker. "Theoretical Studies of Shock-initiated Detonations," Acta Astronautica Vol. 5, pp. 1041-1050.

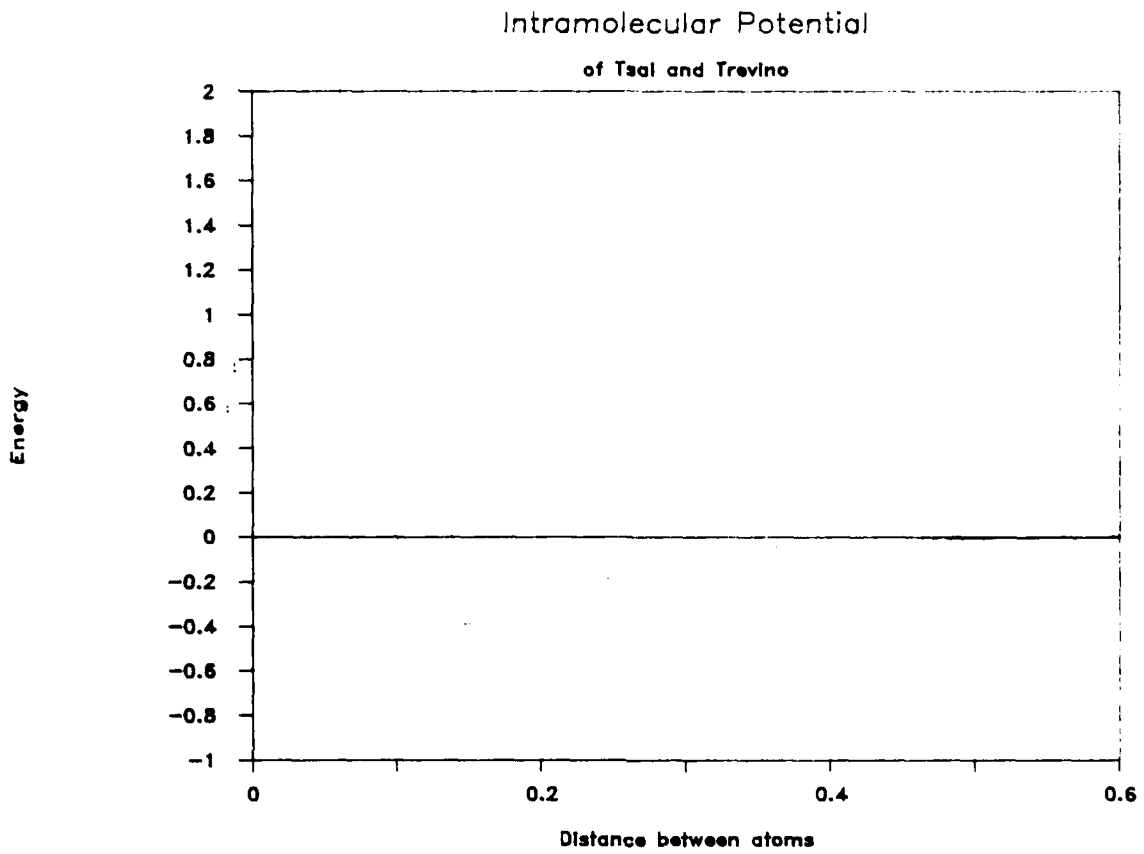
MacDonald, R. A.; and D. H. Tsai. "Molecular Dynamical Calculations of Energy Transport in Crystalline Solids," North Holland Publishing Co. Amsterdam.

Tsai, D. H.; and S. F. Trevino. "Simulation of the Initiation of Detonation in an Energetic Molecular Crystal," J. Chem Phys 81 (12) pt I., 15 Dec 84, pp. 5636-5637.

Acknowledgments

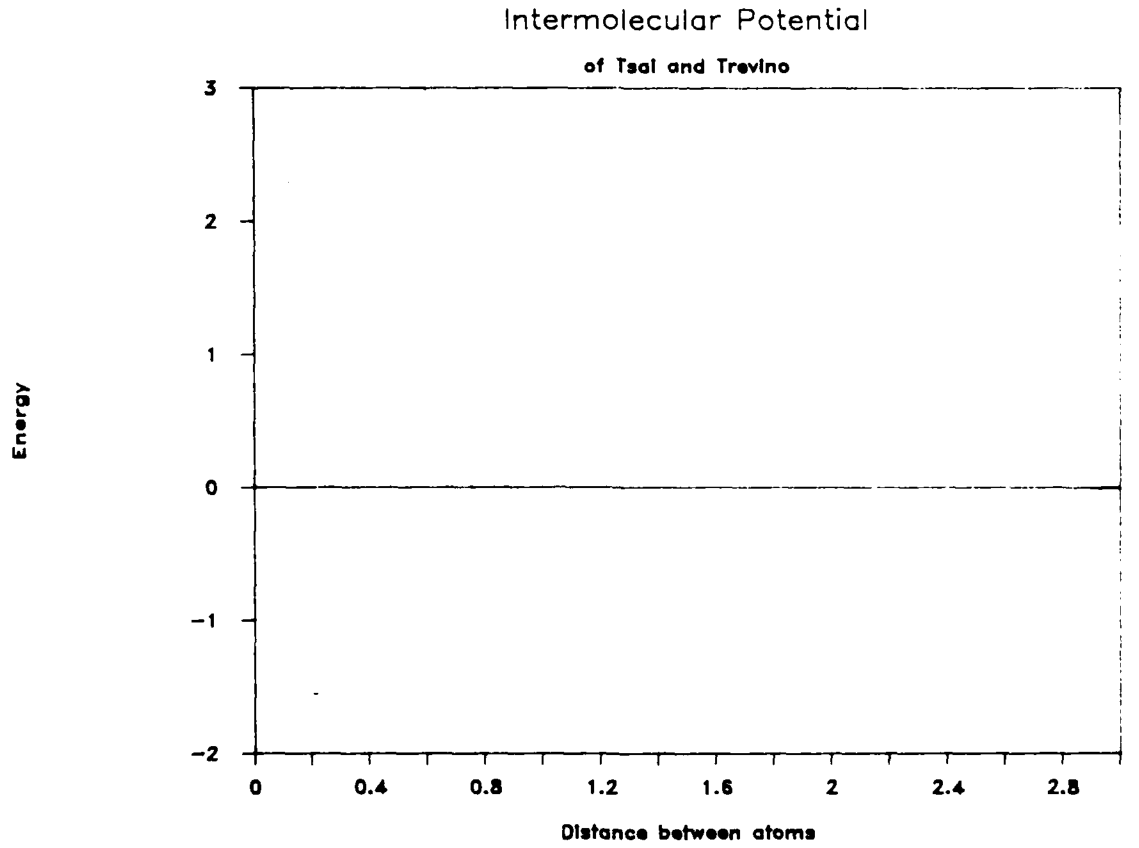
I would like to thank Mr. Steve Satterfield for his help with the computer graphic displays. He also wrote the program to animate the individual frames. The animation makes a fine movie, and it presents the data in an interesting format.

I also want to thank my advisor Associate Professor Mark L. Elert for his patience during the year. His encouragement helped me solve several bugs in the computer program. We learned the fickleness of the CADIG computer, and overcame all mistyped commands to finish with a working program.


Intramolecular Potential

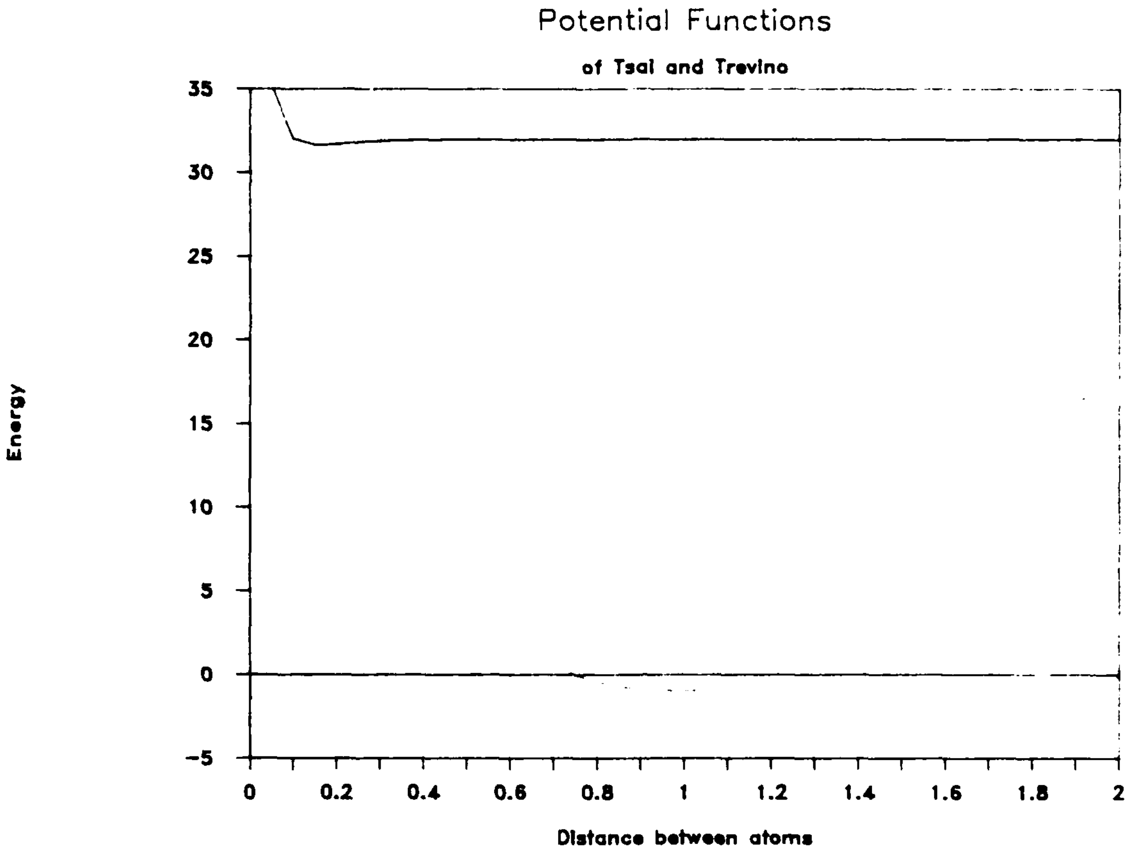
of Tsai and Trevino

Graph 1

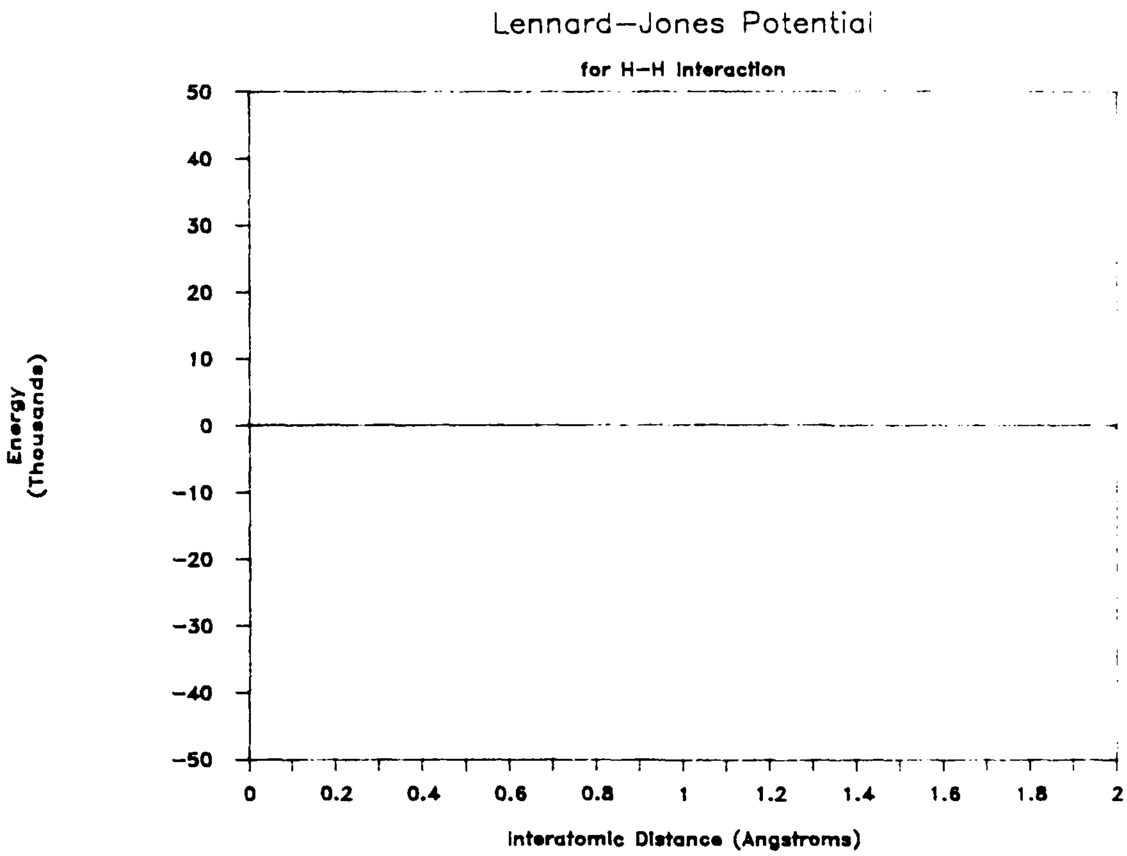

Intermolecular Potential

of Tsai and Trevino

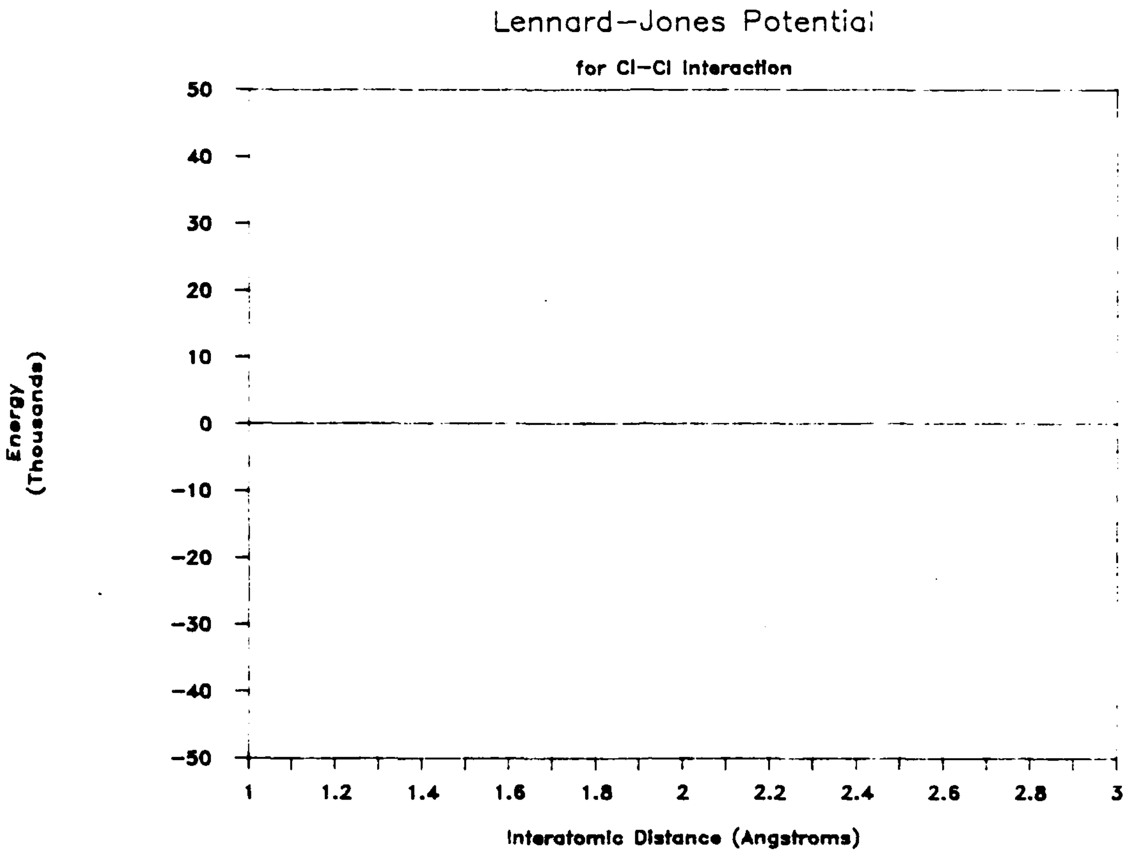
Graph 2


Potential Functions
of Tsai and Trevino
35
30
25
20
15
10
5
0
-5
Energy
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Distance between atoms

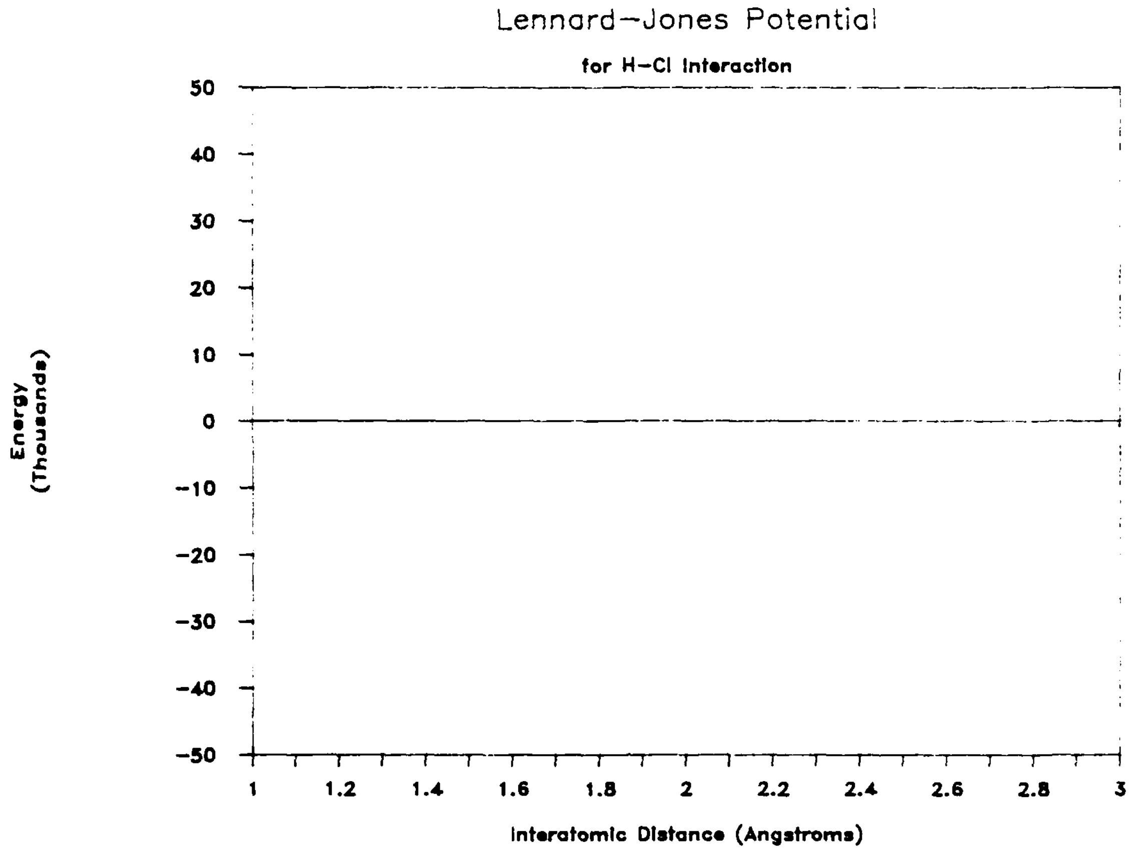
Graph 3


Lennard-Jones Potential
for H-H Interaction
Energy (Thousands)
50
40
30
20
10
0
-10
-20
-30
-40
-50
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Interatomic Distance (Angstroms)

Graph 4


Lennard-Jones Potential
for Cl-Cl Interaction
Energy (Thousands)
50
40
30
20
10
0
-10
-20
-30
-40
-50
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3
Interatomic Distance (Angstroms)

Graph 5


Lennard-Jones Potential
for H-Cl Interaction
Energy (Thousands)
50
40
30
20
10
0
-10
-20
-30
-40
-50
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3
Interatomic Distance (Angstroms)

Graph 6

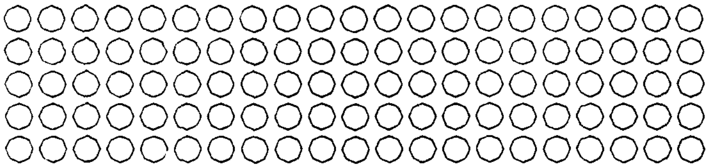


Figure 1

Note that monoatomic molecules are used for simplicity of showing the initial parameters

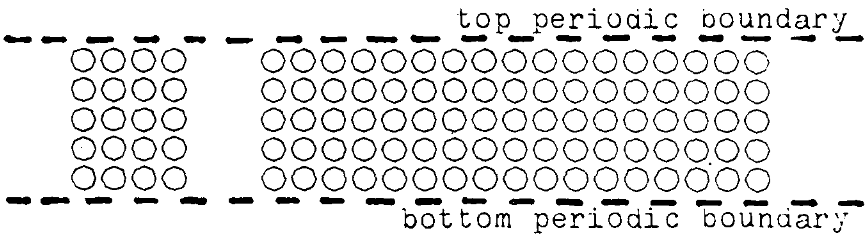


Figure 2

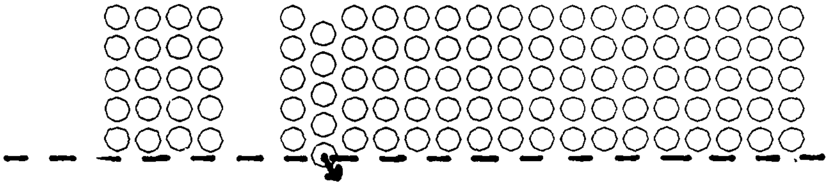


Figure 3

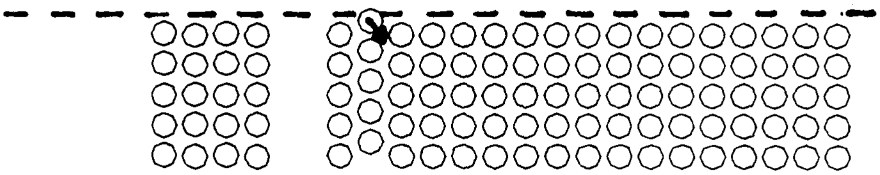


Figure 4

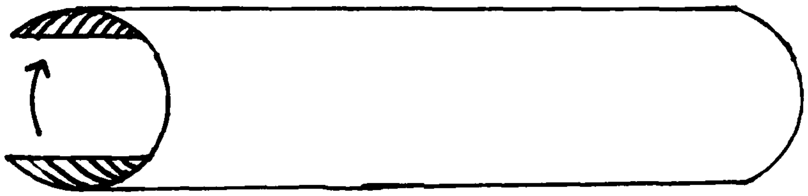


Figure 5

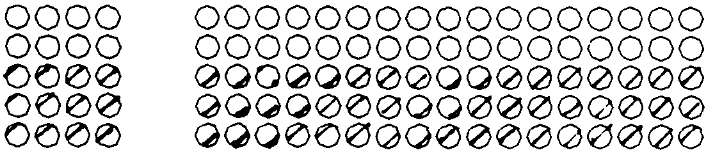


Figure 6

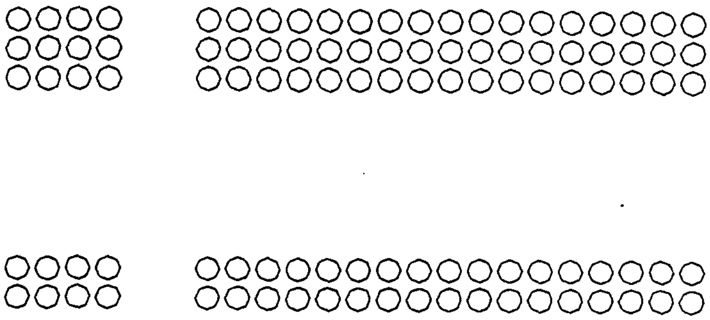


Figure 7

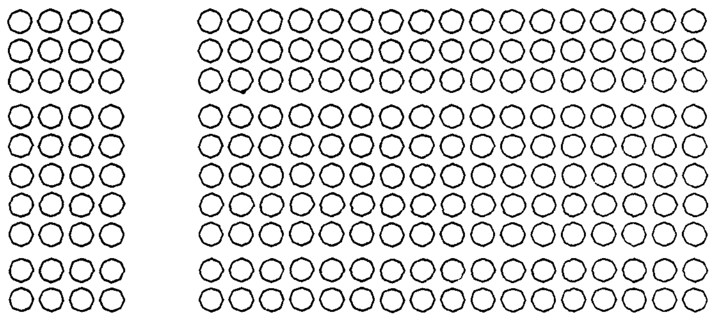


Figure 8

Note that the space in between the new atoms and the original crystal was artificially put so that the difference is easily noticed. During the actual simulation the new expanded crystal is one continous crystal twice the size of the original crystal.

Modified Euler Method

at 0.0000 picoseconds

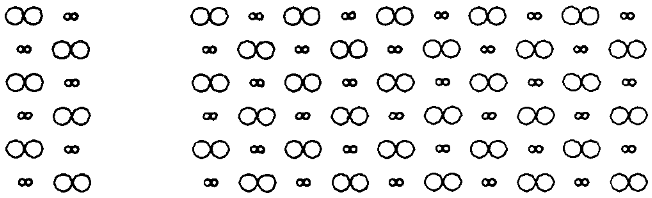


Figure 9

Modified Euler Method

at 0.0999 picoseconds

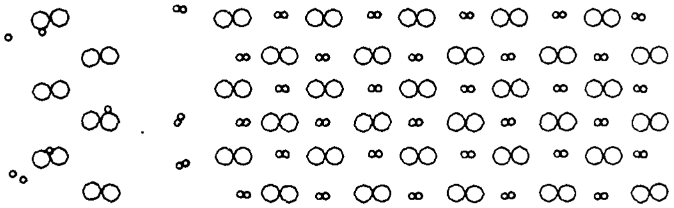


Figure 10

```
C*********************************************************************
C**                                                                 **
C**                                                                 **
C**               LENNARD-JONES (6-12) POTENTIAL                    **
C**          WITH HH & ClCl MOLECULES IN TWO DIMENSIONS             **
C**                                                                 **
C**                             by                                  **
C**                        Alan D. Boyd                             **
C**                    Midshipman First Class                       **
C**                                                                 **
C**                      Trident Project '86                        **
C**                                                                 **
C*********************************************************************
C
C
C                   DIMENSION ALL ARRAYS REQUIRED
C
C
      DIMENSION X(288),Y(288)
      DIMENSION XMOM(144),YMOM(144)
      DIMENSION XVEL(144),YVEL(144)
      DIMENSION XMOMD(144),YMOMD(144)
      DIMENSION X2(288),Y2(288)
      DIMENSION XMOM2(144),YMOM2(144)
      DIMENSION XVEL2(144),YVEL2(144)
      DIMENSION XMOMD2(144),YMOMD2(144)
      DIMENSION T(144)
C
C
C*********************************************************************
C**                                                                 **
C**                                                                 **
C**              SET UP ALL COMMON VARIABLES                        **
C**                                                                 **
C**                                                                 **
C*********************************************************************
C
C
      COMMON/B1/ORIGMOM
C
C
C*********************************************************************
C**                                                                 **
C**              SET UP PARAMETERS FOR CRYSTAL SIZE                 **
C**                                                                 **
C*********************************************************************
C
C
      PARAMETER (XORIGIN = 15.)
C   Check value for sizes and equation for ymove
      PARAMETER (LENGTH = 10.11411)
      PARAMETER (SIZES = 3.69)
      PARAMETER (YMOVE = 6.*SIZES)
C
C
C THE NUMBER OF ATOMS IN THE (ORIGINAL) CRYSTAL IS = N
```

```
      PARAMETER (N = 144)
C
C
C*********************************************************************
C**                                                                 **
C**                   OPEN FILES FOR OUTPUT                         **
C**                                                                 **
C**            FIRST ONE IS FOR CALCULATION OF TOTAL ENERGY         **
C**            SECOND ONE IS FOR STORAGE IN CASE OF SHUTDOWN        **
C**            THE REST ARE FOR STORING THE ATOMS' POSITIONS        **
C**               AT VARIOUS INCREMENTS OF TFINAL                   **
C**                                                                 **
C*********************************************************************
C
C
      OPEN(UNIT=1,STATUS="UNKNOWN",FILE="totena")
      OPEN(UNIT=2,STATUS="UNKNOWN",FILE="savfla")
      OPEN(UNIT=3 ,STATUS="UNKNOWN",FILE="ta0.300")
      OPEN(UNIT=4 ,STATUS="UNKNOWN",FILE="ia1.300")
      OPEN(UNIT=10,STATUS="UNKNOWN",FILE="ta1.300")
      OPEN(UNIT=11,STATUS="UNKNOWN",FILE="ta2.300")
      OPEN(UNIT=12,STATUS="UNKNOWN",FILE="ta3.300")
      OPEN(UNIT=13,STATUS="UNKNOWN",FILE="ta4.300")
      OPEN(UNIT=14,STATUS="UNKNOWN",FILE="ta5.300")
      OPEN(UNIT=15,STATUS="UNKNOWN",FILE="ta6.300")
      OPEN(UNIT=16,STATUS="UNKNOWN",FILE="ta7.300")
      OPEN(UNIT=17,STATUS="UNKNOWN",FILE="ta8.300")
      OPEN(UNIT=18,STATUS="UNKNOWN",FILE="ta9.300")
      OPEN(UNIT=19,STATUS="UNKNOWN",FILE="ta10.300")
C
C
C*********************************************************************
C**                                                                 **
C**         CALL SUBROUTINES TO SET UP ORIGINAL POSITIONS           **
C**                   AND ORIGINAL MOMENTUMS                        **
C**                                                                 **
C*********************************************************************
      CALL POS(X,Y,T)
      ORIGMOM = 100.
      CALL MOM(XMOM,YMOM,T)
C
C
C*********************************************************************
C**                                                                 **
C**          WRITE INFORMATION ABOUT PROGRAM PARAMETERS             **
C**                TO THE TOTAL ENERGY FILES                        **
C**                                                                 **
C*********************************************************************
C
C
      WRITE(1,100)
  100 FORMAT(1X,"Modified Euler Method   tinc = .0001 tfinal = .1")
C
C
C
C
```

```
C********************************************************************
C**                                                                **
C**                        UNITS                                   **
C**                                                                **
C**                                                                **
C**            Mass = atomic mass units    (amu)                   **
C**                                                                **
C**            Distance = Angstrom    (A)                          **
C**                                                                **
C**            Time = picosecond    (psec)                         **
C**                                                                **
C**    THESE UNITS MAKE THE NUMBERS MANAGABLE IN TERMS             **
C**                OF THE EXPONENTS                                **
C**                                                                **
C********************************************************************
C
C
C psec
      TINIT = 0.0
      TINC = 0.0001
      TFINAL = 0.1
C
C
C
C
C
C
C********************************************************************
C**                                                                **
C**          CALL SUBROUTINE TO FIND TOTAL ENERGY OF               **
C**               SYSTEM AT INITIAL CONDITIONS                     **
C**                                                                **
C********************************************************************
C
C
      CALL TOTEN(X,Y,XMOM,YMOM,T)
C
C
C "DIVIDE" IS USED TO BREAK THE OUTPUT INTO "DIVIDE" NUMBER OF
C       OUTPUT FILES
C
C
      DIVIDE = 10.
C
C
C********************************************************************
C**                                                                **
C**             SET THE INITIAL TIME, FINAL TIME,                  **
C**             AND TIME INCREMENT FOR THE PROGRAM                 **
C**                                                                **
C********************************************************************
C
C
      DO 110 TIM = TINIT,TFINAL,TINC
C
C
```

```
C*********************************************************************
C**                                                                 **
C** CALL SUBROUTINES TO FIND VELOCITIES AND MOMENTUM DERIVATIVES    **
C**                                                                 **
C*********************************************************************
C
C
      IT = 4
      CALL WRITFILE(X,Y,T,IT,TIM)
      CALL LJ(X,Y,XMOM,YMOM,XVEL,YVEL,XMOMD,YMOMD,T)
C
C
C*********************************************************************
C**                                                                 **
C** USE EULER METHOD TO FIND TEMPORARY NEW POSITIONS AND MOMENTUMS **
C**                                                                 **
C*********************************************************************
C
C
      DO 120 I = 1,N,1
      X2(I)  = X(I)  + TINC * XVEL(I)
      Y2(I)  = Y(I)  + TINC * YVEL(I)
      XMOM2(I)  = XMOM(I)  + TINC * XMOMD(I)
      YMOM2(I)  = YMOM(I)  + TINC *YMOMD(I)
  120 CONTINUE
C
C
C*********************************************************************
C**                                                                 **
C**       CALL SUBROUTINES TO FIND VELOCITIES AND MOMENTUM          **
C**                      DERIVATIVES USING                          **
C**  TEMPORARY POSITIONS AND MOMENTUMS FOR USE IN MODIFIED EULER    **
C**                          METHOD                                 **
C**                                                                 **
C*********************************************************************
C
C
      CALL LJ  (X2,Y2,XMOM2,YMOM2,XVEL2,YVEL2,XMOMD2,YMOMD2,T)
C
C
C*********************************************************************
C**                                                                 **
C** USE MODIFIED EULER METHOD TO FIND ACTUAL NEW POSITIONS AND      **
C**                          METHODS                                **
C**                                                                 **
C*********************************************************************
C
C
C SAVE STEPS BY USING TINC2 = TINC / 2.
C
C
      TINC2 = TINC / 2.
C
C
      DO 130 I = 1,N,1
      X(I)  = X(I)  + TINC2 *  (XVEL(I)  + XVEL2(I))
```

```
      Y(I) = Y(I) + TINC2 * (YVEL(I) + YVEL2(I))
      XMOM(I) = XMOM(I) + TINC2 * (XMOMD(I) + XMOMD2(I))
      YMOM(I) = YMOM(I) + TINC2 * (YMOMD(I) + YMOMD2(I))
  130 CONTINUE
C
C
C CHECK TO SEE IF  TIM = ONE OF THE DESIRED OUTPUT FLAGS
C
C
      DO 140 KONT = 1,10,1
      CNT = FLOAT(KONT)
      IF (ABS(TIM-(CNT*TFINAL/DIVIDE)).LT.1E-6) THEN
      I = KONT + 9
      CALL WRITFILE(X,Y,T,I,TIM)
      CALL TOTEN(X,Y,XMOM,YMOM,T)
      CALL STORFILE(TIM,X,Y,XMOM,YMOM,T)
      ENDIF
  140 CONTINUE
C
C
C MOVE ATOM IF IT HAS GONE OUTSIDE OF ORIGINAL CRYSTAL PARAMETERS
C
C
      DO 150 K = 1,N,1
      IF (Y(K).GT.YMOVE)     Y(K)   = Y(K)  - YMOVE
      IF (Y(K).LT.0.)  Y(K)  = Y(K)  + YMOVE
      IF (X(K).LT.0.) THEN
      XMOM(K)  = -XMOM(K)
      X(K)  = - X(K)
      ENDIF
  150 CONTINUE
  110 CONTINUE
C CALCULATE THE TOTAL ENERGY OF THE SYSTEM AT THE END OF THE PROGRAM
      CALL TOTEN(X,Y,XMOM,YMOM,T)
      END
C
C
      SUBROUTINE LJ(X,Y,XMOM,YMOM,XVEL,YVEL,XMOMD,YMOMD,T)
C
C
      DIMENSION X(*),Y(*),XMOM(*),YMOM(*)
      DIMENSION XVEL(*),YVEL(*),XMOMD(*),YMOMD(*)
      DIMENSION T(*)
      PARAMETER (N = 144)
      PARAMETER (XORIGIN = 15.)
      PARAMETER (SIZES = 3.69)
      PARAMETER (YMOVE = 6.*SIZES)
C
C COORDINATE DERIVATIVES
C
      DO 200 I = 1,N,1
C REM TO REZERO XMOMD AND YMOMD
      XMOMD(I) = 0.
      YMOMD(I) = 0.
C FIND THE VELOCITY FOR EACH ATOM
      IF(T(I).LT.3.5) THEN
```

```
      AMASS = 1.008
      ELSE
      AMASS = 35.45
      ENDIF
      YVEL(I) = YMOM(I) / AMASS
  200 XVEL(I) = XMOM(I)/ AMASS
C ENLARGE THE CRYSTAL TO GET STABILITY FOR THE MONITERED ATOMS
      DO 210 I = 1,N,1
      IF (Y(I).GT. 0.5 * YMOVE) THEN
      Y(N + I) = Y(I) - YMOVE
      X(N + I) = X(I)
      ELSE
      Y(N + I) = Y(I) + YMOVE
      X(N + I) = X(I)
      ENDIF
  210 CONTINUE
C I HAVE SET UP MOLECULES ABOVE AND BELOW THE INITIAL CRYSTAL
C THEN GOING THROUGH THE LIST ONCE CALCULATE ALL OF THE FORCES
      CALL AL(X,Y,XMOM,YMOM,XVEL,YVEL,XMOMD,YMOMD,T)
      RETURN
      END
C
C
      SUBROUTINE AL(X,Y,XMOM,YMOM,XVEL,YVEL,XMOMD,YMOMD,T)
      DIMENSION X(*),Y(*),XMOM(*),YMOM(*)
      DIMENSION XVEL(*),YVEL(*),XMOMD(*),YMOMD(*)
      DIMENSION T(*)
      PARAMETER (N = 144)
      PARAMETER (SIZES = 3.69)
      PARAMETER (YMOVE = 6.*SIZES)
C
C INTERMEDIATE FUNCTIONS
C
      DO 300 I = 1,N,1
      DO 310 J = 1,2*N ,1
C FIND THE DISTANCE BETWEEN EACH SET OF TWO MOLECULES
C       in EACH COORDINATE DIRECTION (X & Y)
      IF (I.EQ.J) GO TO 310
      RX = X(I) - X(J)
      RY = Y(I) - Y(J)
C FIND TOTAL DISTANCE BETWEEN EACH SET OF TWO MOLECULES
      R2 = RX * RX + RY * RY
      R6 = R2 * R2 * R2
      R = SQRT(R2)
C FOR LARGE VALUE OF R THE FORCE EXERTED BY MOLECULES IS SMALL
C THEREFORE SAVE TIME BY NOT CALCULATING THE FORCE IF R IS TOO
C   LARGE     (SKIP BY USING IF THEN STATEMENT)
      IF(R.GT.1.E6) GO TO 310
C FIND THE LENNARD-JONES FUNCTION BETWEEN EACH SET OF TWO MOLECULES
C REMEMBER THAT THE MOLECULES EXERT EQUAL AND OPPOSITE FORCES ON ANOTH
      IF(R.LT.0.0005*SIZES) THEN
      WRITE(0,320)
  320 FORMAT("PARTICLES ARE GETTING TOO CLOSE ERROR HAS OCCURRED")
      FPART = 1000.
      ELSE
      IF( (T(I).LT.3.5) .AND. (T(J).LT.3.5) ) THEN
```

```
C A    amu * A**8 / psec**2
       A = 15032.44
C B      amu * A**14 / psec**2
       B = 1296.3691
       ELSE
       IF( (T(I).GT.3.5)  .AND.  (T(J).GT.3.5) )  THEN
       A = 2998181.1
       B = 92505858.
       ELSE
       A = 368924.09
       B = 788575.56
       ENDIF
       ENDIF
       FPART = 6. * ( 2. * B / R6 - A) / (R2*R6)
       ENDIF
C
C   MOMENTUM DERIVATIVES
C
C    THE RX AND RY TERMS ACCOUNT FOR THE DIRECTION OF THE FORCE
       XVDSUB = FPART * RX
       YVDSUB = FPART * RY
C SUM ALL THE INDIVIDUAL FORCES FOR EACH MOLECULE
       XMOMD(I)  = XMOMD(I)  + XVDSUB
       YMOMD(I)  = YMOMD(I)  + YVDSUB
  310 CONTINUE
  300 CONTINUE
       RETURN
       END
       SUBROUTINE MOM(XMOM,YMOM,T)
       DIMENSION XMOM(*),YMOM(*)
       DIMENSION T(*)
       COMMON/B1/ORIGMOM
       PARAMETER  (N = 144)
C SET ORIGINAL MOMENTUMS TO 0.
       DO 400 I = 1,N,1
       XMOM(I)  = 0.
       YMOM(I)  = 0.
  400 CONTINUE
C GIVE FIRST FOUR ATOMS IN EACH ROW INITIAL MOMENTUM
       DO 410 I = 1,6,1
       NSTART = 24 *  (I-1)  + 1
       NEND = NSTART + 3
       DO 420 J = NSTART,NEND,1
       Y = RAND(J)
       Y =  (Y - 0.5)  * 50.
       YMOM(J)  = Y
       IF(T(J).LT.3.5)  THEN
       XMOM(J)  = ORIGMOM
       ELSE
       XMOM(J)  = 35 * ORIGMOM
       ENDIF
  420 CONTINUE
  410 CONTINUE
       RETURN
       END
       SUBROUTINE TOTEN(X,Y,XMOM,YMOM,T)
```

```
      DIMENSION T(*)
      DIMENSION X(*),Y(*)
      DIMENSION XMOM(*),YMOM(*)
      PARAMETER  (N = 144)
      PARAMETER  (SIZES = 3.69)
      PARAMETER  (YMOVE = 6.*SIZES)
      DO 500 I = 1,N,1
      IF  (Y(I).GT. 0.5 * YMOVE) THEN
      Y(N + I)  = Y(I)  - YMOVE
      X(N + I)  = X(I)
      ELSE
      Y(N + I)  = Y(I)  + YMOVE
      X(N + I)  = X(I)
      ENDIF
  500 CONTINUE
      ESYS = 0.
      just = 0
      DO 510 I = 1,N,1
      XYMOMT2 = (XMOM(I)*XMOM(I)  + YMOM(I)*YMOM(I))
      IF(T(I).LT.3.5) THEN
      AMASS = 1.008
      ELSE
      AMASS = 35.45
      ENDIF
      EKINTOT = XYMOMT2 / 2. / AMASS
      FTOT = 0.
      DO 520 J = (I+1),2*N,1
      RX = X(I)  - X(J)
      RY = Y(I)  -Y(J)
      R2 = RX*RX + RY*RY
      R6 = R2 * R2 * R2
      IF  (R6.GT.1.E6) GO TO 520
      IF(J.GT.N) THEN
      J1 = J - N
      ELSE
      J1 = J
      ENDIF
      IF(  (T(I).LT.3.5)  .AND.  (T(J1).LT.3.5)  ) THEN
C A   amu * A**8 / psec**2
      A = 15032.44
C B     amu * A**14 / psec**2
      B = 1296.3691
      ELSE
      IF(  (T(I).GT.3.5)  .AND.  (T(J1).GT.3.5)  ) THEN
      A = 2998181.1
      B = 92505858.
      ELSE
      A = 368924.09
      B = 788575.56
      ENDIF
      ENDIF
C
C
      CALL STORFILE(TIM,X,Y,XMOM,YMOM,T)
      FPART = (B / (R6 * R6)) - (A / R6)
      IF(J.GT.N) FPART = 0.5 * FPART
```

```
      FTOT = FTOT + FPART
  520 CONTINUE
      ESYS = ESYS + EKINTOT + FTOT
  510 CONTINUE
      WRITE(1,530)ESYS
  530 FORMAT(1X,"THE SYSTEM'S ENERGY IS ",E12.4)
      RETURN
      END
      SUBROUTINE STORFILE(TIM,X,Y,XMOM,YMOM,T)
      DIMENSION X(*),Y(*),XMOM(*),YMOM(*)
      DIMENSION T(*)
      PARAMETER (N = 144)
      REWIND 2
      WRITE(2,600)
  600 FORMAT(1X,"FILE CONTAINING TIM,I,T,X,Y,XMOM, and YMOM")
      WRITE(2,610)TIM
  610 FORMAT(1X,F7.4)
      DO 620 I = 1,N,1
      WRITE(2,630)I,T(I),X(I),Y(I),XMOM(I),YMOM(I)
  630 FORMAT(1X,I4,",",F4.1,",",F10.6,",",F10.6,",",F13.6,",",F13.6)
  620 CONTINUE
      RETURN
      END
      SUBROUTINE POS(X,Y,T)
       DIMENSION T(*)
       DIMENSION X(*),Y(*)
C set up array for type of molecule
C*************************************************
C**                                             **
C**                   KEY                       **
C**                                             **
C**           H-H           1                   **
C**            H            2                   **
C**          (H)-Cl         3                   **
C**           H-(Cl)        4                   **
C**            Cl           5                   **
C**           Cl-Cl         6                   **
C**                                             **
C**                                             **
C*************************************************
       PARAMETER (N = 144)
       PARAMETER(XORIGIN = 15.)
       PARAMETER(SIZES = 3.69)
       PARAMETER(LENGTH = 10.11411)
C
C
C
C************************************************
C**                                            **
C**      Set up Original Positions             **
C**                                            **
C**                and                         **
C**                                            **
C**      Label the Molecules by Type           **
C**                                            **
C************************************************
```

```
C********************************************************************
C**                                                                **
C**        DISTHH and DISTCLCL ARE INTRAMOLECULAR DISTANCES        **
C**                                                                **
C**              DISTHCL IS AN INTERMOLECULAR DISTANCE             **
C**                                                                **
C**        DISTCENT IS THE DISTANCE REQUIRED TO ALIGN              **
C**          THE CENTER OF THE BONDS BETWEEN ROWS                  **
C**                                                                **
C**             All distances are in Angstroms                     **
C**                                                                **
C********************************************************************
C
C
      DISTHH = .74611
      DISTHCL = 3.69
      DISTCLCL = 1.988
      DISTCENT = .620945
      K = -3
      J = 0
      DO 700 YPOS = .5*SIZES,6.2*SIZES,SIZES
      DO 710 XPOS = XORIGIN, XORIGIN + 5.1 * LENGTH, LENGTH
      K = K + 4
      IF(J.EQ.0) THEN
      X(K)  = XPOS
      X(K+2)  = XPOS + DISTHH
      X(K+1)  = XPOS + DISTHH + DISTHCL
      X(K+3)  = XPOS + DISTHH + DISTHCL + DISTCLCL
      T(K)  = 1.
      T(K+2)  = 1.
      T(K+1)  = 6.
      T(K+3)  = 6.
      ELSE
      X(K)  = XPOS - DISTCENT
      X(K+2)  = XPOS - DISTCENT + DISTCLCL
      X(K+1)  = XPOS - DISTCENT + DISTCLCL + DISTHCL
      X(K+3)  = XPOS - DISTCENT + DISTCLCL + DISTHCL + DISTHH
      T(K)  = 6.
      T(K+2)  = 6.
      T(K+1)  = 1.
      T(K+3)  = 1.
      ENDIF
      Y(K)  = YPOS
      Y(K+2)  = YPOS
      Y(K+1)  = YPOS
      Y(K+3)  = YPOS
  710 CONTINUE
      IF(J.EQ.0) THEN
      J = 1
      ELSE
      J = 0
      ENDIF
  700 CONTINUE
C MOVE FIRST FOUR ATOMS IN EACH ROW BACK
      DO 410 I = 1,6,1
      NSTART = 24 * (I-1) + 1
```

```
      NEND = NSTART + 3
      DO 420 J = NSTART,NEND,1
      X(J) = X(J) - 5.
  420 CONTINUE
  410 CONTINUE
       I = 3
       CALL WRITEFILE(X,Y,T,I,TIM)
       RETURN
       END
       SUBROUTINE WRITEFILE(X,Y,T,I,TIM)
       DIMENSION X(*),Y(*),T(*)
       PARAMETER (N = 144)
       REWIND I
       WRITE(I,800)
  800  FORMAT(1X,"display world;")
       WRITE(I,810)
       WRITE(I,810)
  810  FORMAT("     ")
       WRITE(I,820)
  820  FORMAT(1X,"     label:=begin_structure")
       WRITE(I,830)
  830  FORMAT(1X,'           character scale 5,5;")
       WRITE(I,840)
  840  FORMAT(1X,"           characters 25,50 'Modified Euler Method';"
       J = I - 9
       WRITE (I,850)TIM
  850  FORMAT(1X,"           characters 25,40 'at ",F6.4," seconds';")
       WRITE(I,860)
  860  FORMAT(1X," end_structure;")
       WRITE(I,810)
       WRITE(I,810)
C
C
       DO 870 J = 1,N,1
       IF (T(J).LT.3.5) RDIS = .37
       IF (T(J).GT.3.5) RDIS = .99
       RADIUS = RDIS * 0.7010678
       PY1 = Y(J) + RDIS
       PX2 = X(J) - RADIUS
       PY2 = Y(J) + RADIUS
       PX3 = X(J) - RDIS
       PX4 = X(J) - RADIUS
       PY4 = Y(J) - RADIUS
       PY5 = Y(J) - RDIS
       PX6 = X(J) + RADIUS
       PY6 = Y(J) - RADIUS
       PX7 = X(J) + RDIS
       PX8 = X(J) + RADIUS
       PY8 = Y(J) + RADIUS
       K1 = J + 100
       WRITE(I,880)K1
  880  FORMAT(1X,"c",I3,":=vector_list n = 20")
       WRITE(I,890)X(J),PY1
       WRITE(I,890)PX2,PY2
       WRITE(I,890)PX3,Y(J)
       WRITE(I,890)PX4,PY4
```

```
      WRITE(I,890)X(J),PY5
      WRITE(I,890)PX6,PY6
      WRITE(I,890)PX7,Y(J)
      WRITE(I,890)PX8,PY8
      WRITE(I,890)X(J),PY1
 890  FORMAT("          ",F9.4,",",F9.4)
      WRITE(I,900)
 900  FORMAT("              ;")
      WRITE(I,910)
      WRITE(I,910)
 910  FORMAT("                 ")
 870  CONTINUE
      WRITE(I,920)
 920  FORMAT(1X,"world:=begin_structure")
      WRITE(I,930)
 930  FORMAT(1X,"   window x=0:150 y=-20:130;")
      DO 940 K = 1,N,1
      K1 = K + 100
      IF  (T(K).EQ.1.)  J = 120
      IF  (T(K).EQ.2.)  J = 168
      IF  (T(K).EQ.3.)  J = 359
      IF  (T(K).EQ.4.)  J = 359
      IF  (T(K).EQ.5.)  J = 216
      IF  (T(K).EQ.6.)  J = 264
      WRITE(I,950)J,K1
 950  FORMAT(1X,"    set color ",I3,",1 applied to c",I3,";")
 940  CONTINUE
      WRITE(I,955)
 955  FORMAT(1X,"      instance of label;")
      WRITE(I,960)
 960  FORMAT(1X," end_structure;")
      RETURN
      END
```

```
      SUBROUTINE RECALF(X,Y,XMOM,YMOM,T,TIMA)
      DIMENSION X(*),Y(*),XMOM(*),YMOM(*)
      DIMENSION T(*)
      PARAMETER (N = 144)
      READ(8,300)TIMA
300   FORMAT(F8.5)
      DO 200 I = 1,N,1
      READ(8,400)A,INT,T(I),C,X(I),B,Y(I),D,XMOM(I),E,YMOM(I)
400   FORMAT(F3.1,I4,2F4.1,F10.6,F3.1,F10.6,F3.1,F13.6,F3.1,F13.6)
200   CONTINUE
      WRITE(0,3)TIMA
  3   FORMAT(1x,f8.5)
      RETURN
      END
```

```
    label0:=begin_structure
          character scale 3,3;
          characters 11,50 'Modified Euler Method';
          characters 11,40 'at 0.0000 picoseconds';
 end_structure;


  c0101:=vector_list n = 20
          5.0000,   2.2150
          4.7406,   2.1044
          4.6300,   1.8450
          4.7406,   1.5856
          5.0000,   1.4750
          5.2594,   1.5856
          5.3700,   1.8450
          5.2594,   2.1044
          5.0000,   2.2150
           ;


  c0102:=vector_list n = 20
          9.4361,   2.8350
          8.7421,   2.5391
          8.4461,   1.8450
          8.7421,   1.1509
          9.4361,   0.8550
         10.1302,   1.1509
         10.4261,   1.8450
         10.1302,   2.5391
          9.4361,   2.8350
           ;


  c0103:=vector_list n = 20
          5.7461,   2.2150
          5.4867,   2.1044
          5.3761,   1.8450
          5.4867,   1.5856
          5.7461,   1.4750
          6.0055,   1.5856
          6.1161,   1.8450
          6.0055,   2.1044
          5.7461,   2.2150
           ;


  c0104:=vector_list n = 20
         11.4241,   2.8350
         10.7301,   2.5391
         10.4341,   1.8450
         10.7301,   1.1509
         11.4241,   0.8550
         12.1182,   1.1509
         12.4141,   1.8450
```

```
               12.1182,   2.5391
               11.4241,   2.8350
                 ;


   c0105:=vector_list n = 20
               25.0000,   2.2150
               24.7406,   2.1044
               24.6300,   1.8450
               24.7406,   1.5856
               25.0000,   1.4750
               25.2594,   1.5856
               25.3700,   1.8450
               25.2594,   2.1044
               25.0000,   2.2150
                 ;


   c0106:=vector_list n = 20
               29.4361,   2.8350
               28.7421,   2.5391
               28.4461,   1.8450
               28.7421,   1.1509
               29.4361,   0.8550
               30.1302,   1.1509
               30.4261,   1.8450
               30.1302,   2.5391
               29.4361,   2.8350
                 ;


   c0107:=vector_list n = 20
               25.7461,   2.2150
               25.4867,   2.1044
               25.3761,   1.8450
               25.4867,   1.5856
               25.7461,   1.4750
               26.0055,   1.5856
               26.1161,   1.8450
               26.0055,   2.1044
               25.7461,   2.2150
                 ;


   c0108:=vector_list n = 20
               31.4241,   2.8350
               30.7301,   2.5391
               30.4341,   1.8450
               30.7301,   1.1509
               31.4241,   0.8550
               32.1182,   1.1509
               32.4141,   1.8450
               32.1182,   2.5391
               31.4241,   2.8350
                 ;
```

```
c0109:=vector_list n = 20
        35.0000,   2.2150
        34.7406,   2.1044
        34.6300,   1.8450
        34.7406,   1.5856
        35.0000,   1.4750
        35.2594,   1.5856
        35.3700,   1.8450
        35.2594,   2.1044
        35.0000,   2.2150
          ;


c0110:=vector_list n = 20
        39.4361,   2.8350
        38.7421,   2.5391
        38.4461,   1.8450
        38.7421,   1.1509
        39.4361,   0.8550
        40.1302,   1.1509
        40.4261,   1.8450
        40.1302,   2.5391
        39.4361,   2.8350
          ;


c0111:=vector_list n = 20
        35.7461,   2.2150
        35.4867,   2.1044
        35.3761,   1.8450
        35.4867,   1.5856
        35.7461,   1.4750
        36.0055,   1.5856
        36.1161,   1.8450
        36.0055,   2.1044
        35.7461,   2.2150
          ;


c0112:=vector_list n = 20
        41.4241,   2.8350
        40.7300,   2.5391
        40.4341,   1.8450
        40.7300,   1.1509
        41.4241,   0.8550
        42.1182,   1.1509
        42.4141,   1.8450
        42.1182,   2.5391
        41.4241,   2.8350
          ;
```

The listings for atoms c0113 to c0243 are omitted to keep the number of pages to a minimium

```
 c0244:=vector_list n = 20
           70.8032,  20.6650
           70.5438,  20.5544
           70.4332,  20.2950
           70.5438,  20.0356
           70.8032,  19.9250
           71.0626,  20.0356
           71.1732,  20.2950
           71.0626,  20.5544
           70.8032,  20.6650
             ;


frame0:=begin_structure
    window x=-30:120 y=-60:90;
     set color 120,1 applied to c0101;
     set color 264,1 applied to c0102;
     set color 120,1 applied to c0103;
     set color 264,1 applied to c0104;
     set color 120,1 applied to c0105;
     set color 264,1 applied to c0106;
     set color 120,1 applied to c0107;
     set color 264,1 applied to c0108;
     set color 120,1 applied to c0109;
     set color 264,1 applied to c0110;
     set color 120,1 applied to c0111;
     set color 264,1 applied to c0112;
     set color 120,1 applied to c0113;
     set color 264,1 applied to c0114;
     set color 120,1 applied to c0115;
     set color 264,1 applied to c0116;
     set color 120,1 applied to c0117;
     set color 264,1 applied to c0118;
     set color 120,1 applied to c0119;
     set color 264,1 applied to c0120;
     set color 120,1 applied to c0121;
     set color 264,1 applied to c0122;
     set color 120,1 applied to c0123;
     set color 264,1 applied to c0124;
```

```
set color 264,1 applied to c0125;
set color 120,1 applied to c0126;
set color 264,1 applied to c0127;
set color 120,1 applied to c0128;
set color 264,1 applied to c0129;
set color 120,1 applied to c0130;
set color 264,1 applied to c0131;
set color 120,1 applied to c0132;
set color 264,1 applied to c0133;
set color 120,1 applied to c0134;
set color 264,1 applied to c0135;
set color 120,1 applied to c0136;
set color 264,1 applied to c0137;
set color 120,1 applied to c0138;
set color 264,1 applied to c0139;
set color 120,1 applied to c0140;
set color 264,1 applied to c0141;
set color 120,1 applied to c0142;
set color 264,1 applied to c0143;
set color 120,1 applied to c0144;
set color 264,1 applied to c0145;
set color 120,1 applied to c0146;
set color 264,1 applied to c0147;
set color 120,1 applied to c0148;
set color 120,1 applied to c0149;
set color 264,1 applied to c0150;
set color 120,1 applied to c0151;
set color 264,1 applied to c0152;
set color 120,1 applied to c0153;
set color 264,1 applied to c0154;
set color 120,1 applied to c0155;
set color 264,1 applied to c0156;
set color 120,1 applied to c0157;
set color 264,1 applied to c0158;
set color 120,1 applied to c0159;
set color 264,1 applied to c0160;
set color 120,1 applied to c0161;
set color 264,1 applied to c0162;
set color 120,1 applied to c0163;
set color 264,1 applied to c0164;
set color 120,1 applied to c0165;
set color 264,1 applied to c0166;
set color 120,1 applied to c0167;
set color 264,1 applied to c0168;
set color 120,1 applied to c0169;
set color 264,1 applied to c0170;
set color 120,1 applied to c0171;
set color 264,1 applied to c0172;
set color 264,1 applied to c0173;
set color 120,1 applied to c0174;
set color 264,1 applied to c0175;
set color 120,1 applied to c0176;
set color 264,1 applied to c0177;
set color 120,1 applied to c0178;
set color 264,1 applied to c0179;
set color 120,1 applied to c0180;
```

```
set color 264,1 applied to c0181;
set color 120,1 applied to c0182;
set color 264,1 applied to c0183;
set color 120,1 applied to c0184;
set color 264,1 applied to c0185;
set color 120,1 applied to c0186;
set color 264,1 applied to c0187;
set color 120,1 applied to c0188;
set color 264,1 applied to c0189;
set color 120,1 applied to c0190;
set color 264,1 applied to c0191;
set color 120,1 applied to c0192;
set color 264,1 applied to c0193;
set color 120,1 applied to c0194;
set color 264,1 applied to c0195;
set color 120,1 applied to c0196;
set color 120,1 applied to c0197;
set color 264,1 applied to c0198;
set color 120,1 applied to c0199;
set color 264,1 applied to c0200;
set color 120,1 applied to c0201;
set color 264,1 applied to c0202;
set color 120,1 applied to c0203;
set color 264,1 applied to c0204;
set color 120,1 applied to c0205;
set color 264,1 applied to c0206;
set color 120,1 applied to c0207;
set color 264,1 applied to c0208;
set color 120,1 applied to c0209;
set color 264,1 applied to c0210;
set color 120,1 applied to c0211;
set color 264,1 applied to c0212;
set color 120,1 applied to c0213;
set color 264,1 applied to c0214;
set color 120,1 applied to c0215;
set color 264,1 applied to c0216;
set color 120,1 applied to c0217;
set color 264,1 applied to c0218;
set color 120,1 applied to c0219;
set color 264,1 applied to c0220;
set color 264,1 applied to c0221;
set color 120,1 applied to c0222;
set color 264,1 applied to c0223;
set color 120,1 applied to c0224;
set color 264,1 applied to c0225;
set color 120,1 applied to c0226;
set color 264,1 applied to c0227;
set color 120,1 applied to c0228;
set color 264,1 applied to c0229;
set color 120,1 applied to c0230;
set color 264,1 applied to c0231;
set color 120,1 applied to c0232;
set color 264,1 applied to c0233;
set color 120,1 applied to c0234;
set color 264,1 applied to c0235;
set color 120,1 applied to c0236;
```

```
     set color 264,1 applied to c0237;
     set color 120,1 applied to c0238;
     set color 264,1 applied to c0239;
     set color 120,1 applied to c0240;
     set color 264,1 applied to c0241;
     set color 120,1 applied to c0242;
     set color 264,1 applied to c0243;
     set color 120,1 applied to c0244;
       instance of label0;
  end_structure;
```

END

10-86

DTIC